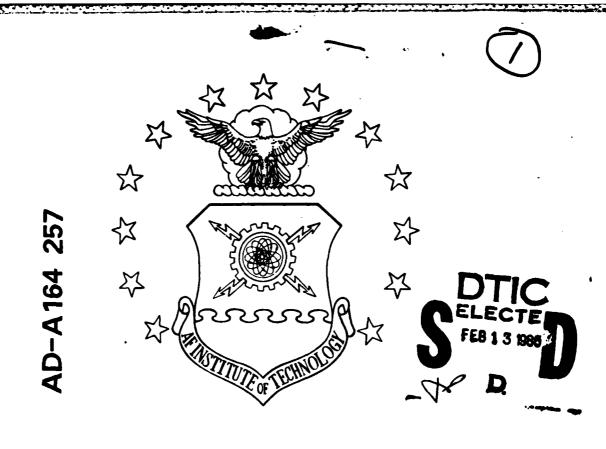
THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 HICROP. (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI. C A BRXLEY DEC 84 AFIT/GC5/ENG/840-2-VOL-2 F/G 9/2 RD-R164 257 1/5 UNCLASSIFIED NL



MICROCOPY RESOLUTION TEST CHART

MATIONAL RUREML OF STANDARDS-1963-A



MODEL OF THE MOTOROLA MC68000 MICROPROCESSOR WITH N.MPC.

THESIS (2 of 3)

Volum

Charles A. Baxley Jr. Captain, USAF

AFIT/GCS/ENG/84D-2-Vol-2

፞ዸጜዼጜ፠ጜ፠፞ጜጜፙፙፙፙቒ፞፞፞ፙ፟ጜኯፘኇ፞ፙኯፙቒቔኇጜዀዀዀዀዀዀዀዄፙፙፙኯኯጛፘጜፘፘቔቔቔኯዹ

FILE COPY

<u>. F</u>

dutainution statement a

Approved for public releases
Distribution Unlimited

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

86 2 12

025



THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 MICROPROCESSOR WITH N.MPC.

THESIS

Charles A. Baxley Jr. affendices A.G.
Captain, USAF

AFIT/GCS/ENG/84D-2-Vol-2

Approved for public release; distribution unlimited

UNCLASSIFIED -

SECUMITY CLASSIFICATION OF THIS PAGE

AD-A164257

REPORT DOCL	MENTATION PAGE	E		
14 REPORT SECURITY CLASSIFICATION UNCLASSIFIED	Ib. RESTRICTIVE M	18. RESTRICTIVE MARKINGS		
26 SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/A	3. DISTRIBUTION/AVAILABILITY OF REPORT		
2b. DE CLASSIFICATION/DOWNGRADING SCHEDULE	1 ' '	Approved for public release;		
	distributi	distribution unlimited.		
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING OR	GANIZATION R	EPORT NUMBER	6)
AFIT/GCS/ENG/84-D-2				
64 NAME OF PERFORMING ORGANIZATION Bb. OFFICE SYMBO	L 74. NAME OF MONIT	TORING ORGAN	IZATION	
School of Engineering AFIT/ENG				
6c. ADDRESS (City, State and ZIP Code)	7b. AODRESS (City,	State and ZIP Coo	le j	
Air Force Institute of Technology	1			- 1
Wright-Patterson AFB, Ohio 45433				Ī
AFSC/FTD SOMEONING BE OFFICE SYMBO (If applicable)	9. PHOCUREMENT I	9. PHOCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
Sc. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUR	NOING NOS	·	
	PROGRAM	PROJECT	TASK	WORK UNIT
1	ELEMENT NO.	NO.	NO.	NO.
11 TITLE Include Security Classification;			{	
See Box 19]	1	1 .
12 MERSUNAL AUTHORIS) (Indries A. Baxley Jr., B.S., Capt,	USAF			
134 TYPE OF REPORT 136. TIME COVERED		14. DATE OF REPORT (Yr. Ma., Day) 18. PAGE COUNT 1984 December 1200		
MS Thesis FROM TO 16 SUPPLEMENTARY NOTATION	1984 Deci	ember	1200	
:				1
•				
	S (Continue on reverse if no Design Langua	cessory and ident	ly by bluck number	ion Sat
	r (ISP'), N.m			
**************************************	MC68000, Mic			•
19 ABSTRACT (Continue on reverse if necessary and identify by block nu				
Title: THE SIMULATION AND ANALYSIS MC68000 MICROPROCESSOR WITH Thesis Chairman: Frederick A. Zapk O DISTRIBUTION/AVAILABILITY OF ABSTRACT VICEASSIFIED/UNLIMITED SAME AS RET CLOTIC USERS	N.mPc	Englered for Strat E. W Does for Re At Force In Stages-Patter	public release: Li DEAVER sourch and Professi stitute of Technology son AFB OM 45417	AW APE SHIPE I Y AUG. 9 / Here Burettpuent I (AIB)
TO NAME OF RESPONSIBLE INDIVIDUAL				
Frederick A. Zapka, Major, USA	22b TELEPHONE N		22c. OFFICE SYMBOL AFIT/ENG	

BECURITY CLASSIFICATION OF THIS PAGE

- 18. Microprocessor Simulation, Microprocessor Analysis, Computer Architecture, Microcomputers, Computerized Simulation, Digital Simulation.
- 19. In a prior thesis project, a functional level model of portions of the Motorola MC68000 microprocessor was developed using signal analysis supported by limited technical data. Representative parts of the instruction set and exception processing structure were modeled with the Computer Design Language (CDL). In this follow-on effort, those CDL models are transformed into equivalent models using ISP', an enhanced vers on of the Instruction Set Processor (ISP) hardware design language. This language transformation enabled the models to be simulated using N.mPc, a VAX 11/780-hosted software package developed specifically to support the design of digital systems. To evaluate the correctness of the of the models, the simulation results are analyzed against signal data gathered with the aid of a logic analyzer during the actual operation of the MC68000 when processing the modeled instructions. The accuracy and completeness of the examined models suggests that this functional approach to microprocessor modeling is a valid

Contents

(Volume I)

	pa ge
ı.	IntroductionI-1
	Background
II.	RequirementsII-1
	Introduction
III.	Intro to the Motorola MC68000 MicroprocessorIII-1
	Introduction
īv.	Introduction to N.mPcIV-1
	Introduction
v.	CDL-to-ISP' Model Transformation
	Introductionv-1

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

	CDL/ISP' Comparison
	Effects of Language Differences
	Registers
	Subregistersv-13
	Buses
	Decoders
	Clock
	Clock & Clock-Cycle Counter RepresentationV-18
	Switch
	Model Changes
	Memory Responses
	EXABUF Size
	ABUS Utilization
	Memory Declaration
	High Impedance RepresentationV-25
	Power-On Sequence
	Additional Simulation ComponentsV-26
	Program Loader
	MC68000 Assembler
VI.	Simulation AnalysisVI-l
	•
	IntroductionVI-1
	Analysis ResultsVI-2
	High Impedance StateVI-3
	Data BusVI-4
	Address BusVI-8
	Individual Instruction Simulation ResultsVI-10
	MOVE.W D1,D2VI-12
	MOVE.W D1, (A1)
	MOVE.L D1,A1VI-15
	MOVE.W D1, (A1) +
	MOVE.W D1,04(A1)
	MOVE.W D1,94(A1,D7)
	MOVE.W D1,\$2004VI-23
	MOVE.W A1,D3VI-25
	MOVE.W (A1),D2VI-26
	MOVE.W (A1)+,D6VI-27
	MOVE.W -(A1), D4
	MOVE.W 04(A1),D1
	MOVE.W 04(Al,D7),D2VI-32
	MOVE.W \$2004,D5
	MOVE.W \$2004,\$2008VI-35
	MOVE.W #\$5555,D1VI-38
	JMP (A0)VI-39
	ADD.W D3,D5VI-40
	BEQ STARTVI-41
	BTST D1, (A1)VI-45
	Illegal Instruction ExceptionVI-46
	Address Error ExceptionVI-50
VII.	Conclusions and RecommendationsVII-1

Introduction
VIII. Analysis of Time Spent on ProjectVIII-
Introduction
BibliographyBIB-1
(Volume II)
Appendix A: Local N.mPc Supplement
Appendix B: CDL/ISP' Declaration SectionsB-1
Appendix C: ISP' Models of MC68000 InstructionsC-1
Appendix D: MC68000 Metamicro DescriptionD-1
Appendix E: MC68000 Linking/Loader DescriptionE-1
Appendix F: Test Routines (Metamicro)F-1
Appendix G: Simulation Control Files
(Volume III)
Appendix H: Simulation OutputH-l
Appendix I: MC68000 CDL ModelsI-l
Appendix J: Logic Analyzer DataJ-1
Appendix K: ISP' Model of the RISC 1K-1
Appendix L: Metamicro Description of the RISC 1L-1
VitaVITA-1

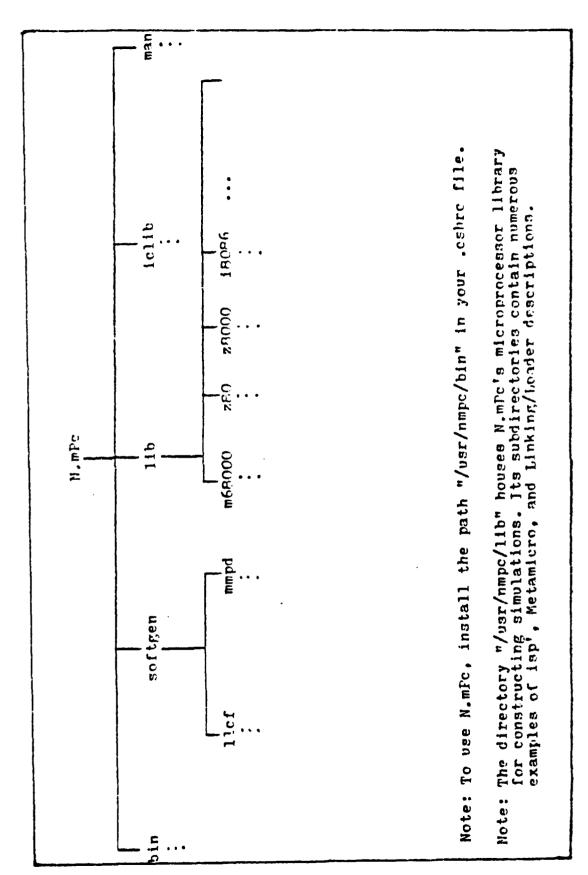


Accesi	on For			
DTIC	ounced			
By				
Availability Codes				
Dist	Avail a:.o Specia	l/or		
A-1	23			

Appendix A: Local N.mPc Supplement

This appendix provides information on the access and use of N.mPc as locally installed on AFIT's SCC VAX 11/780. Included is:

Section	Page
1. Local N.mPc Directory Structure	A-2
2. N.mPc's Library of Microprocessors (nmpc/lib)	A-3
3. N.mPc Organizational Diagram	A-6
4. N.mPc File Names (13:Appendix A)	A-7
5. N.mPc Components (nmpc/man)	A-9
6. VAX N.mPc Post Processor (Case Western Univ)	A-18
7. Sample Post Processor Output	A-22



•

Local M.mPc Directory Structure

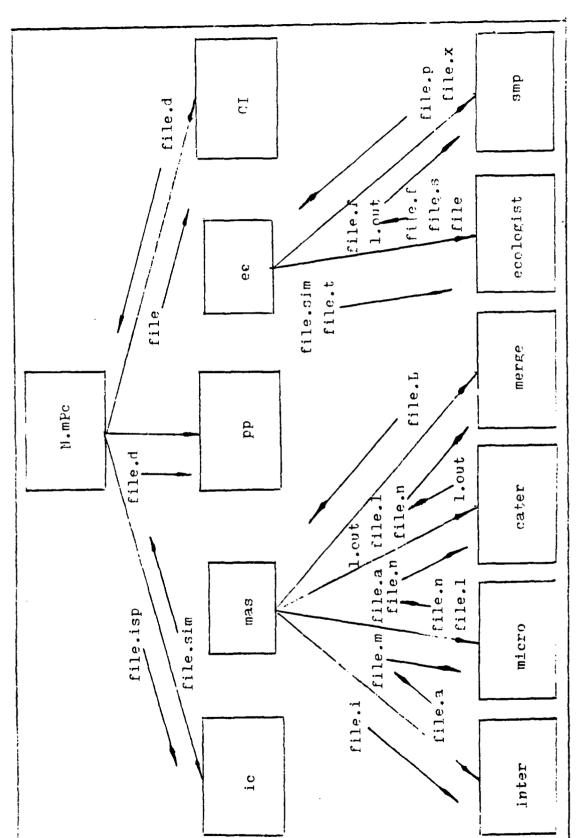
(•

```
r1862a.isp
r1862b.isp
r1862m.isp
                                    nebulas. Isps
nebulas. Isps
nebulas. Isps
.//161
2900/
                                                                                 ./11b/s2658:
                                    ./Itb/nebula/RCS:
bwald1/
bwald2/
18848/
                                    float. Isps.v
                                                                                ./
RCS/
s265#.1
s265#.m
s265#a.1sp
18851/
                                    iek.lsps.v
nebula.lsps.v
 18888/
10#05/
18#86/
m65#2/
                                    nebulas. Isps. v
                                                                                 ./11b/s2658/RCS:
                                    ./!Ib/pdpil:
                                                                                 ./
../
s2658.1,v
s2658.m,v
s2658a.1sp,v
                                    ./
../
RCS/
pdp11.1
pdp11.m
m68888/
nebula/
r 1882/
92658/
19988/
                                                                                 ./11b/t99##:
                                    ./11b/pdp11/RCS:
                                                                                 ./
RCS/
t99##.1
 t1328/
                                    ./,
pdp11.1.v
 z 8#/
 ZBARB/
                                    pdp11.m.v
                                                                                  19988.m
19988a.lsp
 ./11b/a2988:
                                     ./11b/r1882:
19988b. Isp
                                    RCS/
doc/
r18#2.1
                                                                                  ./11b/t9988/RCS:
                                                                                 ./
19988.1.v
                                    r1862.m
r1862a.19p
                                                                                  19988 ....v
 a2911a.19p
                                    r 1892a. s 1m
r 1882c. Isp
                                                                                  19988b. 196. v
  ./116/a2988/ACS1
 ./(16/a2988/0
.//
a2981a.1sp.v
a2989a.1sp.v
a2989a.1sp.v
a2918a.1sp.v
a2918a.1sp.v
                                     r 1802cm. lap
                                                                                  ./11b/t132#:
                                                                                  ./,
                                     ./11b/r1882/RCS:
                                     ://
                                     r1862.1.v
r1862.m.v
r1862e.isp.v
r1862a.sim.v
                                                                                  READ_ME
                                                                                  t1.13p
                                                                                  ti.read
                                                                                  t 17. a
t 17. l
t 17. m
  ./11b/bwaldli
                                     r1882c.19p.v
r1882cm.19p.v
 RCS!
                                                                                  tip.a
tip.i
tip.m
                                      ./11b/r1882/doc+
  bwaldl.a
bwaldl.i
bwaldl.m
                                     :/,
                                                                                  ./11b/t1328/RCS:
                                     cosmac/
                                                                                 ti.isp.v
ti.read.v
tid.i.v
  ./11b/bwaldl/RCS:
                                      ./11b/r1882/doc/cosmac1
  .,,
                                      ...
  bwaldl.a.v
  bwaldi.i.v
bwaldi.m.v
                                      makeum
                                     makeum
r1002
r1802.appx
r1802.ex1
r1802.ex2
r1802.fd
r1802.t
r1802.tc
                                                                                  t14.m.v
t1p.1.v
                                                                                  tip.m.v
   ./11b/bwald2:
                                                                                  ./IIb/vaxII:
                                                                                  ./
ŘČS/
```

```
18060.1
10080.m
18280e.1sp
                                                                     m6582.1sp
m6582.m
10bus.lsp
RCS/
bwald2.a
                                                                      ./11b/m6582/RCS:
                                                                     ./
m6582.1.v
m6582.1sp.v
m6582.as,v
bwald2.1
bwald2.1sp
bwald2.m
bwald2.t
                                  ./11b/18#8#/RCS:
                                  :/,
                                  16488.1.V
                                  1888.m.v
clock. (sp
                                  18880e. isp.v
mem. 15p
                                                                      ./11b/m688#:
                                  ./116/16885:
                                                                     ./
RCS/
m6888.1
./lib/bwald2/RCS:
                                 ./| TD/ 1000.
./
RCS/
1805.m
1805.m
18055.m
1./
10bus.lsp.v
bwald2.a.v
bwald2.lsp.v
bwald2.m.v
bwald2.t.v
clock.lsp.v
mem.lsp.v
                                                                      m6889.m
                                                                      ./11b/m6888/RCS:
                                                                      ...
                                   ./11b/18885/RCS:
                                                                     m6868.1.v
                                 ./
18895.1.v
18895.1.v
18895.m.v
18895a.1sp.v
18895b.1sp.v
                                                                      m6888.m.v
                                                                      ./11b/m688881
 ./116/18848:
                                                                     ./,
RCS/
                                                                      68888.1
 18848a.1
                                   ./11b/18#86:
                                                                      68888 . m
 18848a.1sp
18848a.m
                                                                     68888.n
RCS/
e68886.lsp
                                  8886.1
 ./11b/18848/RCS:
 18848a.1.v
18848a.1.v
18848a.1sp,v
                                                                      e68888.sim
                                  8886.m
                                                                      e68888m. 1sp
                                  8886.n
RCS/
                                                                      e68888m.sim
                                  euwB6max, isp
                                                                      160#88.s1m
                                  e8886min.isp
                                                                      m68888.a
                                                                      m68888.1
 ./115/18851:
                                  e8#86min.sim
 ./
RCS/
dutam.tsp
                                                                      m68888.m
                                                                     m688888 . 1 sp
m688888 . 3 sp
m68888 b . 1 sp
                                   18#86max.sim
                                  18086min.sim
min86mem.isp
 18#51.1
18#51.1sp
18#51.m
18#51.t
                                                                      m68888b.sim
                                                                      m688886bm. 1sp
                                  min8bmem.sim
                                                                      m688886m.sim
                                  read_me
                                                                      read_me
                                  ./11b/18Ø86/RCS:
 progm. isp
                                                                      ./11b/m68###/RCS:
 test
                                                                      68888.1,
                                  ./
8#86.1,v
8#86.m,v
e8#86max.isp,v
e8#86min.isp,v
18#86min.isp,v
  ./11b/18Ø51/RCS:
datam.isp.v
18851.i.v
18851.isp.v
18851.m.v
18851.t.v
                                                                      68888.m.v
e68888.lsp.v
                                                                      e68888m.1sp.v
                                  min86mem. isp,v
                                                                      ./lib/nebula:
                                                                      ./
RCS/
 progm.isp.v
                                   ./11b/m6582:
                                  ./
../
RCS/
m6582.1
  ./116/1608#1
                                                                      float.laps
 i./
RCS/
                                                                      lex. Isps
```

•

...



•

N.mFc (rganization

File Names

Many different file names are involved in a simulation, and it was considered desirable to summerize them in one place.

- xxxx.m A ".m" file is the source input file to the metaMicro assembler. A successful assembly produces a corresponding "xxxx.n" file.
- xxxx.n The intermediate file produced by the metaMicro assembler. Used by the Linking/Loader Allocator.
- yyyy, i The source input to the Linking/Loader Interpreter. Contains the specification of the address resolution process for a given machine.
- yyyy. a The output of the Linking/Loader Interpreter. This file is used by the Allocator to direct the address resolution process.
- 1. out The output of the Allocator. Contains a real machine core image, suitable for simulation after processing by the simulated memory processor.
- zzzz.isp The input to the ISP' compiler, contains ISP' source code.
- zzzz.sim The output of the ISP' compiler, corresponds to the zzzz.isp source input.
- root.t The topology file, the ecologist will build a program called "root", which will be the executable simulation.
- root.s. A symbol table file created by the ecologist, used by the runtime package. "root" is the simulation name.
- root.f The memory list file, produced by the ecologist, and used by the simulated memory processor. Contains the names of all memories used in a simulation.
- wwww.p A processed file created by "smp". Corresponds to a previous Linking/Loader output file which has been renamed. Used by the simulation program and the Simulated Memory Editor. One for each non raw

memory in a simulation.

- root. z Another output of the sup. Contains the global symbols passed on from metaMicro. One per simula-tion.
- root, d Simulation data file, produced by the simulation itself, contains data to be processed by the N. mPc post processor.

(· · ·

Hamle Components

NAME

outer - Linking Alouder Allowior

SYNOPSIS

cour (-options) (u.a (de.) (de2.1 ... (de10.1)

DISCRIPTION

Cater is the second program of the Linking Loader. Cater is driven by a user description of the address resolution process produced by the Interpreter.

Up to 10 files produced by metallitero may be linked at one time.

The description file produced by the interpreter has a "la" extension and the metallicro output files have a "in" extension.

The Allocator produces a file with the name hout. This file can be merged with metallicro listing files with the merge program. A dump of this file may be obtained from the 'indump' program.

OPTIONS

- print the "Logical Space Map", relating logical addresses (produced with the -p option of micro) to physical addresses as established by outer.
- P print the "Physical Space Map", relating physical addresses to logical addresses. The "Physical Space Map" is the inverse of the "Logical Space Map".
- E print the "External Symbol Table", relating external symbols identified by the global label declaration in metall icro to physical addresses where allocated by cater.
- Print the 'Tree Space After Allocation' report. This report shows the unused blocks of the target machine memory space as defined by the space declaration in the Linuxing Loader command program.
- This report shows where transfers have been placed into the target maxime code to maintain logical contiguity of physically disjoint segments of target maxime code.
- R print the "Allocation Order" report, showing the order in which target maxime instructions were allocated.
- A print all reports noted above.
- F. print all physical addresses and complete instructions in hexideomal.
- D print all physical addresses in decimal. Complete instructions are printed in octal.
- O print all physical addresses and complete instructions in octal.
- m allocate nodal fires according to the Modified First Fit allocation algorithm.
- f slocute notal files according to the Fragmentee Memory allocation algorithm
- illocate node ties according to the Low Packang allocation algorithm
- h allocate notal files according to the high Pasking allocation algorithm
- print information, messages that detail the state of the allocation
- s suppress creation of the Lout core image file.
- undicate that the next option (in the form of -value) should be interpreted as the value to might to unablocated memory in the core image file.
- o have the Allocator prompt the user for the target machine address space. The default values are those specified in the "space" declaration of the Interpreter produced file "file.a". The Allocator will attempt to allocate code only in the specified regions.
- t perform a statement trace of the allocation process. This option is useful when debug-

FILES

file.a: parsed file produced by the interpreter file.in: nodal output file produced by metalliaro

i.out · output file of allocator

SEE ALSO

inter(nmpc), mas(nmpc), micro(nupc), mdump(nmpc), Liniang/Loader User's Manual, Version 1.1

NAME

ec - ecologist and simp control program

SYMOPSIS

ec [-upuons] [- upuons] ruot name upuona airustory

DESCRIPTION

ec lets a user run the two simulation creation programs through one controlling program. Options beginning with a '-' are sent to the ecologist, while those beginning with a '+' go to smp. It is also possible to have ec run 'nimake', the NimPo version of 'make'.

The ecologist is driven by a 'topology' file describing the hardware and software components of a particular simulation. The ecologist itself loads the needed ISP modules, and, if no errors occurred, creates a memory list file to drive smp, smp examilies the memories involved and creates memory irriage files in a formal suitable for simulation. Because simulation preparation is decomposed into two parts, it is possible to modify a simulation by only changing the component updated, either hardware through, the ecologist, or software through the smp.

When ed is run, it esists the user if the software, hardware, or both are to be charged. If only nardware is to be charged ed runs the ecologist. If only software is to be charged, only strp is run. Changing both implies running both the ecologist and strp. If errors occurred in running the ecologist, strp will not be run. If the user does not want to be prompted, the 'H', 'S, or 'B' option may be included in the '-' options going to the ecologist. 'H' implies changing only hardware, 'S software, and 'B', both.

The simulation topology file is in a file 'root_name.t'. If a simulation is created through eq. the executable simulation will reside in a file called 'root_name', with no extensions.

Because of the file naming convention, it is most convenient to place a simulation in its own directory.

The 'option directory' is passed to the ecologist, and is used to specify other places to find ISP compiled files. If an 'L' is encountered in the '-' options, the other directory is set to the nimpousp' library directory.

OPTIONS

(

Opuons prefaced by a '-' go to the ecologist, and those prefaced with a '+' go to smp. To run 'nimake' over the topology file before it is given to the ecologist, and the 'M' option to the ecologist options (-M). The other two nimake options ('V' and 'U') can then be added to the invocation options for the ecologist, and they will be sent to nimake, if the 'M' option is also present.

MLKS

root_name.t : topology input file

root_name.f : memory list file produced by the ecologist

root_name.x : global label and memory list file produced by the smp

root_name: the executable simulation /nmpc/icub_ISP abrary directory

SECAUSO

ecologist(nmpc), smp(nmpc), nmake(nmpc), Ecologist User's Manual, Version 1.0

BUCS

Only one instance of '-' or '+' options may occur.

NAME

(,)

ecologist - nmpc simulation topology parser

SYNOPSIS

ecologist [-options] root_name [other_dir]

DESCRIPTION

The ecologist parses a file containing a description of the hardware and software items that make up a given simulation. The hardware modules are ISP output files, and the software modules are core images created by the Linking/Loader. The specified hardware modules are loaded into a program containing a kernel program which creates a complete simulation program. A list of the memory images specified is placed into a file which the Simulated Memory Processor (smp) may examine to process the named memores for simulation.

The file which describes the hardware and software arrangement is called the topology file. Each simulation has a root_name. Severa' files exist which have the root_name, with vanous extensions. The topology file has the root_name with a '.t' extension. Because of the naming conventions used, it is usually desirable to place each simulation in its own directory.

If no errors were encountered in parsing the topology file, the executable program will be in a file with the root name and no extensions.

The option other_dur on the invocation line is used in front of specified ISP output file names as a mechanism to facilitate keeping ISP hardware modules in library directiones.

OPTIONS

- list the topology input as it is being persed.
- s list runtime symbol file.
- m list memory files used.
- t do not remove temporary files (debugging option).
- build a simulation which runs in separate instruction and data space (for 11/45, 55, and 70 CPU's).
- f try and link simulation with one loader call; speeds up the simulation building process If using this option results in a loader error message, do not use it (!).
- 1-9 link in an alternate nunture kernel. The single character 1 to 9 will be appended to the kernel name (kernela), forming a new kernel name. This is useful in environments in which there are multiple kernels, each used for different purposes. Specifying a new kernel which does not exist will result in a loader error.

FILES

root_name: executable simulation program root_name.t: topology file for simulation

root_name.f : memory list file

root_name.x : smp output file with memory names and global labels

rooi_name.s: symbol table

/nmpc/bin/kemel.a: runtime kemel archive

SEE ALSO

ed(nmpc), id(nmpc), smp(nmpc), Ecologist User's Manual

BUCS

NAMES.

このことにはなる。自然などのなるない。

.

ic - ISP Compiler

SYNOISIS

ic [-options] name isp

DESCRIPTION

ic parses source programs in the ISP language producing PDP-11 code to be run under the N.mPc runtime kernel.

Each source file must be terminated with a ".isp" extension. The compiler will produce an output file with the same root name, but with a ".sm" extension.

OPTIONS

- generale a listing.
- p parse only (generate no code)
- s essembly listing of code generated in normal listing.
- t print table of ISP structures used
- w suppress warning messages.
- T turn on trace option.

PILKS

name.isp: ISP source file name name.sm: ISP output file name

/nmpc/bin/libisp.a: ISP runtime library

SEE ALSO

ecologist(nmpc), nmake(nmpc), 1SP User's Manual

BUGS

Only one source program may be compiled at a time.

NAME

inter - Linking Loader Command Program Interpreter

SYNOPSIS

inter [-options] file.i

DESCRIPTION

Inter is the name of the Linking/Loader Command Program Interpreter, a program which translates a user description of the address resolution process for a purticular machine. Interproduces a file with the same root name as the input file, but with a "la" extension. This "la" file drives the Allocator (cater), the program which actually links metall icro nodal output files, producing executable core image files.

OPTIONS

- force a listing of the command program.
 - invoke the "1" option and produce a listing of included files.
- s suppress the creation of the ".a" file (syntax pass only).
- force a listing of the contents of the command program and number statements in the mode and transfer sections. This option is most useful when using the allocator statement trace facility (-t).

PILES

roctl: input file to be filtered through the C preprocessor, making the root; file.

root: : input file to be interpreted.

roota: interpreted file to be input to the Alincator.

SEE ALSO

cater(nmpc), mas(nmpc), micro(nmpc), Linking Loader User's Manual, Version 1.1

BUCS

HAME

mas - Miloro A Ssembler, metaly loro, Linking /Louder, Milerge Interface

SIRIOAYZ

mas [[-][+]opuoris] [machine] [files.m] [files.n] [files.one]

DESCRIPTION

Through mas, a user may execute metal/licro, the Interpreter, the Allocator, and Merge, the four software generation components of N.mPc. The destination of the options depends on the interited use of mas.

To Assemble Programs

If mas is to be used to run only metallicro, options should begin with "-" and an arbitrary number of ".m" or metallicro source flies may be specified. One of the options that must be used is the "-c" option which, as in the C compaler, instructs the software to produce only object flies (".n"). Assembling 3 flies to produce 3 nodal (object) files would appear as

mas -cXXX fuel.m fue2 m fue3 m

XXX are other required metall icro options.

To interpret Files

M as can be used to run the interpreter by using the following format:

mas [-options] machine

The options here are sent to the Interpreter. Machine is the name of the Interpreter input file, without the "..." extension

Max will examine the directory where Linking/Loader machine descriptions are stored for a file with the name "machine a". If this file is found, it is assumed to be the current Interpreter output, and has terminates.

If the file cannot be found, mas constructs the name "machine.l" and attempts to find that file. If this file can be found, the C preprocessor is used to filter the file into another file named "machine.i" If the "triacnine.l" file cannot be found. C preprocessor filtering is not performed and no errors are produced.

M as then constructs the name "machine,i" and sends this name to the Interpreter to be interpreted.

The directory where the Limong/Loader command files are stored is installation dependent, but can be overridden with the "-l" option. For installations running Version 7 Unix, the shell variable "IPATE" can be set and exported. Mas will use "IPATE"s value unless overridden with "-l". Also, the shell variable "MACHINE" can be set and exported and mas will use its value as the "machine" name argument noted above.

To Link Files

M as can be called to run the Allocator and produce an Lout file. The format is

mas [+ options] machine file1.n [file2.n ... file10.n]

The "+" options are sent to the Allocator. The "machine" name argument is treated as above in the section entitled "To Interpret Files".

If one argument is specified with a "core" extension, the Lout file is renamed "file.core".

Mas can be used in combinations of the above descriptions. For example, consider a core image which is built from three metallicro programs. Assume two are current and one has been charged. Mas can be used to run metallicro over the modified file, and then call the Allocator to perform the linking. Format:

NAME

merge - Merge metall icro listing files with A liocator core image files

SYNOPSIS

merge [-options] [files core]

DESCRIPTION

merge merges listing files created by metall icro with the information contained in the core image file produced by the Allocator, producing final listing files.

The Allocator builds the core image file with all information necessary to access all files required in final listing file production.

OPTIONS

- display physical addresses in decimal and instruction values in octal
- X display physical addresses and instruction values in hendecimal.
- O display physical addresses and instruction values in octal. This is the default radix

USACE

Merge merges all core images files specified as arguments (or Lout if no files are specified). The merging operation involves

- 1. Parsing listing files (file.1) produced by metaMicro.
- 2. Extracting logical addresses imbedded in those files (added to a metal/icro listing by specifying -p) and mapping to physical addressing (using the logical to physical map imbedded in the core image file).
- 3. Determining the length of the associated instruction (by reading the file n nodal output file produced by metall icro).
- 4. Extracting the final instruction value (from the core image file specified as an argument or liout by default), and
- 5. Adding the physical address and instruction value to the listing file to produce the final listing file (file.L).

PILES

lout: default core image file if no .core files are specified.
file.1: listing files produced by metaMicro with a -p option.

file.n: nodal files produced by metal icro. file.l: final listing files produced by merge.

SEB ALSO

micro(nmpc), cater(nmpc), mas(nmpc), metaM icro-Linking/Loader Utilities User's Manual, Version 2.0

DIACNOSTICS

Designed (hopefully) to be self-explanatory. Merge really gets confused if one of the listing files, nodal files, or core image file is out of date with the other files. The error messages produced don't make sense, start all over with metall icro and the Allocator before merging again.

BUCS

NAMB

micro - metal/ icro A ssembler

SYNOPSIS

micro [-options] file m

DESCRIPTION

micro is the metallicro assembler, a general microprocessor assembler which is user programmed for each assembly, metallicro output files have the same name as the input file, except that the ".m" extension is changed to a ".n" extension. Notal files produced by metallicro may be link/loaded by cater, a generalized linking/loader.

OPTIONS

- generate a source listing, with decimal line numbers
- invoke the "1" option and list the source code in included files.
- e envoke the "I" option and expand macros in the instruction section.
- E invoke the 'l' option and expand macros in both the declaration and instruction sections.
- I invoke the "I" option and snow how "II" statements cause text redirection.
- p place option. Invoke the "I" option and display the logical address of each instruction in the file. This option is required if "merge" is going to postprocess metall icro source files.
- a invoke the 'l", "e", 'i", and 'f" options.

USACE

metal/ icro's output should be redirected to a file named 'file.!" for merge to postprocess.

FILES

file.m: metall icro input file.

file.n: metall icro object code output file.

file.l: metaMicro listing file, for merge. /tmp/microXXXXX: temporary file for macros.

SEE ALSO

mas(nmpc), inter(nmpc), cater(nmpc), merge(nmpc),

metal icro User's Manual, Version 3.1

Linking/Loader User's Manual, Version 1.1

metal/ icro-Linlong/Loader Utilities User's Manual, Version 2.0

BUGS

Only one file may be assembled per invocation.

NAME

smp - Simulated M emory Processor

SYNOPSIS

smp [-options] memory_list.f

DESCRIPTION

smp takes core image files produced by the allocator (l.out) and prepares them for simulation by creating fixed size pages suitable for swapping at simulation time. It also collects globally defined labels into a common file to be used by the simulation and the Simulated Memory Editor (sme).

smp is driven by a memory list file produced by the ecologist. This file contains a list of the memones to be used in the given simulation. This file has the root name of the simulation, with a '.f' extension.

smp processes each file mentioned in the memory list file, leaving the output in a file with the same name as the input, except for the added '.p' extension. This '.p' file is used by the simulation

The file containing the global symbols found in each memory list file has the simulation root name, with a '.x' extension.

smp must be run after each change in simulation software, smp may be run directly, or through the 'ec' program.

OPTIONS

- generate a listing of smp's actions.
- g generate a listing including global labels found in each memory.
- He labels shown in the gloption have addresses in hex.
- D label addresses are in decimal (default is octal)
- a page size is 32 words
- b page size is 64 words.
- c page size is 128 words.
- d page size is 256 words (default size).
- e page size is 512 words.
- f page size is 1024 words.
- s user specied page size (format sXXX, XXX is a number from 0 to 5000, must be last option).

FILES

rool_name.f : memory list file

root_name.x: processed memory list with global labels

lout : file produced by linking loader allocator

flie_name.p : flie processed by smp

SEE ALSO

ed nmpc), mas(nmpc), ecologist(nmpc), Ecologists User's Manual, Version 1.0

BUGS

A Simple Vax N.mPc Post Processor (V1.7)

Greg Ordy

Dept. of Computer Engr. and Science Case Western Reserve University Cleveland, Ohio 44106

ABSTRACT

This manual describes a simple post processor for the VAX implementations of the N.mPc System. It will run under both the VMS and Unix (4.1 BSD) operating systems. This post processor is considerably simpler than the one written for the PDP-11 implementations, however it performs the basic function of reading the data files (*.d) produced by the simulation, and displaying its contents. The post processor is written in the C language, as is most of N.mPc.

1. Introduction

(

The VAX post processor is named 'pp', as is the PDP-ll version. When running a simulation, the trace command can be used to generate time-tagged data which is placed in a file with the root simulation name, and a '.d' suffix. Each unique trace command is tagged with a monitor number which is the number returned when the trace command is executed. For example, here is a fragment from an example simulation:

...
trace cpu:ir
monitor number 3
...

In response to the trace command request the command interpreter named that monitor '#3'. That number (3) is also used to refer to that traced data stream in the post processor.

When running the post processor, any set, or all of the data traces may be displayed. In addition, the output base may be set to any one of either 2, 8, 10, or 16. Note that in all cases simulation time is always displayed in base 10.

Because this program was quickly designed to be a temporary replacement for a more powerful post processor, there are some limitations.

- 1) Any structure which is wider than 32 bits cannot be displayed as a decimal number.
- 2) Simulation time is only maintained to a precision of 32 bits.

Even with these restrictions, 'pp' should be quite useful.

2. Post Processor Options

The name of the post processor is 'pp'. Each invocation must have the name of the trace data file to be examined specified. Note that 'pp' will accept names with or without the '.d' suffix. In addition to the data file name, one or more program options may be specified.

In addition to the basic function of displaying trace data, the post processor summerizes the data found in the file in terms of the number of entries of each monitor. When the post processor is finished displaying data, a summary will be displayed.

2.1. Number Base Selection

The default output base for all data values is base 16. Options to modify this are:

- -b Display data in base 2 (binary).
- Display data in base 8 (octal).
- -d Display data in base 10 (decimal).
- -x Display data in base 16 (hex).

For bases other than 2, all leading zeros are removed from data values. Base 2 data is shown at the full field size, rounded up to the next multiple of 8 bits.

2.2. Monitor Specification

COCOCOLO PERCENTA PERCENTA DE LA COCOCOCA DE LA COCOCA DE LA COCOCA DE LA COCOCA DE LA COCA DEL LA COCA DEL LA COCA DE LA COCA DE LA COCA DE LA COCA DEL LA COCA DE LA COCA DE LA COCA DE LA COCA DEL LA COCADA DEL

By default, all monitors encountered in the trace file are displayed. The set may be reduced by explicitly specifying the monitors to be shown. Each monitor number is specified as '#x', where 'x' is the monitor number. If any monitor numbers are specified, all unspecified monitors are automatically disregarded.

2.3. Other Options

The post processor also supports the following options:

- -s Only generate a summary, no data display.
- -n Do not generate a summary.
- option is most useful on VMS systems. Note that error information will still be sent to the terminal. The file name is directly concatenated to the 'f' option character. For example, if you want the output to go into a file named 'data.txt', you would specify to 'pp':

pp -fdata.txt

Other options may not be used in the same option string as the '-f' option, after the file name.

3. Misc.

If no options are specified, the default is to display all monitors in base 16, as well as the data summary. Displayed time values, as well as data in the summary section is always displayed in decimal.

Example:

pp -d siml

Display all monitors in the data file 'siml.d' in base 10. A summary will be printed at the end.

Example:

pp -b #3 #12 try.d

Display data for only monitors 3 and 12 in file 'try.d' in base 2. A summary will be printed at the end.

Simulation time is displayed whenever it changes relative to the display of monitors, and to determine the time that a displayed value occurred, just look backwards to find the last time entry.

Due to the implementation of the runtime command interpreter, names of <u>display</u> commands will also be placed in the trace data file. Their data, however, is shown on the terminal, and is not placed in the file.

Currently, all monitor data is shown in the same number base. The internal structure of the program is set up to allow a separate base per monitor. To implement per monitor display bases all that need be changed is the portion of the program which deals with parsing program options.

(

Cample Post Processor (utput

```
created Fri Jul 13 12:56:33 1984 output to: simdata
All monitors displayed.
Monitors displayed in base 2.
at t .
                                      at t = 8
                                       at t = B
            trace :K_W' = 00000001
'trace :AS' = 00000001
'trace :K_W' = 00000001
   6.0
            'trace :FC' = 88888018
            t = 120
            # 1 B
            t = 42#
              trace :LDS' = 00800001
            'trace :LDS' = ชองพระส
'trace :UDS' = ชองพระส
'trace mem:DATA read' = ฮสอธเ
'trace mem:DATA read' = ฮสอธเ
'trace mem:DIACK' = ฮฮิลพ์พระฮ
                                      6406868666668686
         129
           .trace :D) read. - คนกลิทคยอกพอดอกคอดอกอกอกอกอก

;trace :bC read. = กกรุษทองอณอดพลอดอกอกอกอนอบกลได้

;trace :bC read. = กกรุษทองอณอดพลอดอกอกอดอนอบกลได้

;trace :bC = กกรุษทองอนอกอนอดอกอกอดอกอดอกอก

;trace :bC = กกรุษทองอน
            #j 'trace (D[2] read' #
         #24 'trace :R_W' = 80000881
```

```
# 2 B
  418
   t = 960
  = 1828
    'trace :FC' = #8888888
  426
   - 88488881818181818181818181818181
- 1888
  #24 'trace :R_W' = 80808881
= 1148
  = 1500
  t = 156#
    'trace :FC' *
  #26
         BRUPBORY
  #24 'trace :R_W' = 888888881
t = 1688
   #26 'trace :FC' = #8880018
#14 'trace :ADDRESS' = #88
  #26
          t = 1748
    #28
   # 1 B
   #16
t = 2648
    :FC' - 20000000
   #26
```

TOTAL PROPERTIES A SECURITY

(

```
= 2160
                  424 'trace :R_W' - 888888881
      2228
                            'trace :FC' - MBBBBBBBB
                            414
     2280
                  #18 'trace :LDS' = #86660000
#16 'trace :UDS' = #86660000
#16 'trace :AS' = #86660000
                  2589
                            trace mem:DIACK, = 88999889, trace mem:DIACK, = 88989889, sepages | trace :AS, = 889888888 | trace :AS, = 88988888 | trace :AS, = 8898888 | trace :AS, = 88988888 | trace :AS, = 88888888 | trace :AS, = 8888888 | trace :AS, = 888888 | trace :AS, = 8888888 | trace :AS, = 888888 | trace :AS, = 8888888 | trace :AS, = 888888 | trace :AS, = 888888
                  #28
                  418
      2648
                  - DBACGOOBBOABOOBBOAT BIBIBIBIBIBIBI
   - 2788
                   #24 'trace :R_W' - 88888881
        2768
                  = 2828
                   #28 'trace .LDS' - Memousass
                   #18 'trace :UUS' - 066011868
#16 'trace :AS' - 066011868
#29 'trace mem:DATA read' - 0611618868060806001
#29 'trace mem:DATA read' - 061161886806080001
   - 312B
                            - 3188
                  trace :SR read; = BDBB100088888888188
                    •6
                   424 'trace :R_W' = BUBUDBU!
         3300
                   3368
                            'trace :LDS' = BOUUDBOR
'trace :UUS' = BUBBUUBB
'trace :AS' = BUBBUUBB
'trace mem:DATA read' = BOUUBUBUIIBIBIBBB
'trace mem:DATA read' = BIBUIIIBIBIBBBB
'trace mem:DTACK' = BBBBBBBI
                  # 2 H
                   # 1 B
       3668
                  3728
```

```
- 3788
                    #24 'trace :R_W' = MBBBBBB
       3848
                               'trace :FC' = ##800018
'trace :AUDRESS' = 00000000000180000000000
                    #26
     3966
                                 'trace :LDS' = 80300088
'trace :UDS' = 80000888
'trace :AS' = 8000088
                    #28
                               'trace mem:DATA read' = Budusananasul
'trace mem:DATA read' = Budusananasul
'trace mem:DIACK' = ubusanasi
      4200
                               'trace :LUS' = 000000000
'trace :UNS' = 00000001
'trace :AS' = 40000001
'trace mem:DATA read' = 000000000000000
'trace mem:UTACK' = 000000000
                    # 2 H
     4260
                                 'trace :FC' = 00000000
                    #26
                               trace :ADURESS - DRITHINGROUNNOOL
                    4328
                    #24 'trace :R_W' = BOBUBBB!
                                'trace :FC' - 88888818
                    #26
                                 'trace :ADDRESS' . D38800608008188008008081
                     414
               4 4
                                'trace : LUS' = 860618886'
'trace : AS' = 8886818886'
'trace : AS' = 8886818888
                    428
                     # L B
                               4748
                                'trace ilDS' = BURDDOD!
'trace iAS' = BURDDOD!
'trace mem:DATA read' = RESE!
'trace mem:DATA read' = RESE!
                     #18
                                                                                                             • Z 3
                                    trace (FC' - 88866088
                    #26
                    #6 'trace iSR read' = อกกลุ่มของอุดอลายด
#6 'trace iSR read' = อกกลุ่มของอุดอลายด
#6 'trace iSR read' = อกกลุ่มของอุดอลายด
                      4868
                     #24 'trace :R_W' - DBBBBBBB
                    4988
                    #ZB 'trace :[DG' = Buffilluf
#IB 'trace :UDS' = Buffilluf
#IB 'trace :AS' = Buffilluf
#Z9 'trace mem:DATA read' = Buffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffilluffil
                                'trace :LDS' = 80000001
'trace :UUS' = 80000001
'trace :AS' = 80000001
'trace mem:DATA read' = 8000
'trace mem:DIACK' = 80000000
                    #2H
                     #1B
                                                                                                        - 06088898888888
```

```
= 5488
         #24 'trace :R_W' = 808008#1
         *56 ,flace : LC, = 90003010
t - 5528
             'trace iLDS' = UUJUUUUJ
'trace iUDS' = UUJUUUJU
'trace iAS' = UUJUUUJU
'trace mem:UATA read' = BUJUUUJUUJUUJUU
'trace mem:OATA read' = 60!!UJUUJUUJUUJUU
'trace mem:DTACK' = UUJUUJU
         428
         #18
         #16
#39
         ₽Ž3
  - 5828
         #53 'trace mem:DTACK' = NOWBOOR

#10 'trace :Y2' = NOWRORD

#10 'trace :Y2' = NOWRORD

#10 'trace mem:DTACK' = NOWBOORD

#23 'trace mem:DTACK' = NOWBOORD

#23 'trace mem:DTACK' = NOWBOORD

#23 'trace mem:DTACK' = NOWBOORD
    5886
         #24 'trace :R_W' = 88800881
              6868
         6360
              'trace :LDS' = BB000001
'trace :UDS' = 00000001
'trace :AS' = 00000001
'trace mem:DATA read' = 900000000
'trace mem:DTACK' = 88000000
         #28
          #16
    6428
         #24 'trace :R_W' = BEBBBBBI
              .14
t = 6600
              'trace :LDS' = UBUGREGE
'trace :LDS' = BEBUUUE
'trace :AS' = BEBUUUE
'trace mem:DATA read' = BEGEEBHEBEBEBEBEBE
'trace mem:DATA read' = DEIIBIMBEBEBEBEBE
'trace mem:DTACK' = WEMBEBEE
'trace mem:DTACK' = WEMBEBEE
```

(

■ かっているのは、●だっていている。●でいるのではないのでは、●だっているではない。

• •

Appendix B: ISP*/CDL Declaration Sections

```
1*
                                                    */
               MC68000 ISP' Model Structure Declarations
/*
                                                    X/
state
/*
                                                    */
            M68000 Programming Registers
/*
                                                    */
DE0:73:31:0>,
                    ! 8 Data Registers
A[0:5](31:0),
                    ! 7 Address Registers
UA7 (31:0),
                    ! User Stack Pointer
SA7<31:0>,
                    ! System Stack Pointer
PC(31:0).
                    ! Frogram Counter
SR<15:0%,
                    ! Status Register
/×
                                                    */
/¥
            Temporary Internal Registers
                                                    */
/x
PFR<15:0>,
                     ! Prefetch Register
IR<15:0>.
                     ! Instruction Register
FC<2:0>,
                    ! Function Code Register
EXDBUF (15:0).
                    ! External Data Bus Buffer Register
                     ! External Address Bus Buffer Register(changed)
EXABUF<23:10.
ALUBUF1 (31:0),
                    ! ALU Buffer 1
                    ! ALU Buffer 2
ALUBUF2<31:0>,
GTEMP<15:0>,
                    ! Temporary Data Storage
DISPEG<31:0>,
                    ! Temporary Displacement Storage
SETEMP 15:00.
                    ! Temporary Status Register Storage
                     ! (Exception Processing)
IRTEMP<15:0>,
                    ! Temporary Instruction Register Storage
                     ! (Exception Frocessing)
TEMPADR<31:0>,
                     ! Temporary Cycle Address Storage
                     ! (Exception Processing)
ACTYPE<15:0>,
                     ! Temporary Access Type Storage
                     ! (Exception Processing)
VECADR<23:0>,
                     ! Temporary Vector Address Storage
                     ! (Exception Processing)
HANADR (31:0).
                     ! Temporary Address Storage For
                     ! Exception Handler Routine
```

```
T<7:0>.
                        ! Clock Cycle Counter
RESET.
                     ! Reset Flip-Flop
HALT,
                      ! Halt Flip-Flop
                      ! Read/Write Flip-Flop
FiW,
                     ! Address Bus Buffer Enable
ADENABLE.
DIBENABLE,
                     ! Data Bus Buffer Enable
ASN,
                     ! Address Strobe Flip-Flop
LDSN,
                     ! Lower Nata Strobe Flip-Flop
UDSN.
                     ! Upper Data Strobe Flip-Flop
DITACKN,
                      ! Data Transfer Acknowledge Flip-Flop
COUT,
                     ! Carry Flip-Flop
EXCEPT.
                     ! Exception Processing Flip-Flop
READY.
                     ! Ready Flip-Flop
/x
                                                              */
/x
       Model transformation modifications:
                                                              */
/*
                                                              */
/ X
           1) CDL decoder structure nonexistent in ISF' and un-
                                                             */
/*
       necessary for model. Eliminated.
                                                              */
/x

 Multi-phase clock structure nonexistent in ISP'.

                                                              */
/x
       Operations on registers will provide its equivalent.
                                                              x/
           3) Switch structure nonexistent in ISP'. Operation on a
/*
                                                              */
/*
       register will provide its equivalent.
                                                              */
/*
           4) The declared bus structures are modeled with registers */
/*
       without loss of model accurracy. This done to maintain model
/*
       equivalency and simplicity.
                                                              */
/*
           5) The memory word length was reduced from 16 to 8 bit
       words to coincide with the ECR's 32-Kbyte memory, to agree with*/
/*
/x
       their PC incrementation, and to enable the use of existing
                                                             x/
       MC68000 assembler and linker/loader models. The memory was
/*
                                                             */
/*
       also reduced from 8 Mwords to 32 Kbytes.
                                                             1/
                                                             */
/*
IABUS<31:0>.
                        ! Internal Address Bus
IDBUS<31:0>,
                        ! Internal Data Bus
SWITCH.
                     ! Power Switch
PHI1.
                     ! Phase 1 Of Two-Phase Clock
                     ! Phase 2 Of Two-Phase Clock
PHI2;
port
/*
                                                              11/
/×
                                                              */
             External Address and Data Bus
/*
                                                              */
! Data Bus
DBUS(15:0).
ABUS<23:1>;
                        ! Address Rus
format
```

```
/*
                                                            */
/*
                 Register Subfields
                                                            */
                                                            */
/#
! Program Counter Address Field
FCADUR
          = PC\langle 23:0\rangle,
SRTRACE
          = SR<15>,
                       ! Trace Bit
SEMODE
          = SR<13>,
                       ! Mode Selection Bit
SRCARRY
          = SR<0>,
                       ! Carry Bit
                       ! Overflow Bit
SROVER
          = SR<1>,
SRZERO
          = SR(2),
                       ! Zero Bit
SRNEG
                       ! Negative Bit
          = SR<3>,
SREX
          = SR<4>,
                       ! Extend Bit
SRMASK
          = SR<10:8>,
                       ! Interrupt Mask
FCSFACE
          = FC<1:0>,
                       ! Memory Access Address Space
FCMODE
          = FC\langle 2\rangle,
                       ! User/Supervisor Mode Bit
F'CLUW
          = PC<15:0>,
                       ! PC Low Word
PCHI
          = PC(31116),
                       ! PC High Word
                       ! DEOJ Low Word
IIOLWORI
          = In[0](15;0),
DILWORD
          = DE13<15:0>.
                       ! D[1] Low Word
II2LWORD
          = D[2](15:0),
                       ! II[2] Low Word
D3LWORD
          = D[3](15:0),
                       ! D[3] Low Word
II4LWORE
          = D[43<15:0>,
                       ! B[4] Low Word
DSLWORD
          = DE53<15:0>,
                       ! DE53 Low Word
II6LWORI
          = DE63<15:0>,
                       ! DE63 Low Word
D7LWORD
          = DE73<15:0>.
                       ! BC73 Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW
          = HANADR<15:0>, ! HANADR Low Word
HANADRHI
          = HANADR<31:16>,! HANADR High Word
TEMPAURLOW = TEMPAUR<15:0>,! TEMPAUR Low Word
          = TEMPADR<31:16>;! TEMPADR High Word
TEMPADRHI
MEMOTY
/*
                                                            */
                 16K 16-Bit Word Internal Memory
                                                            */
/*
                                                            */
/x
```

ME0:327673<7:0>;

"al Reclarations

```
$ Programming registers
              D0(0-31),
                                    $data registers
Register,
              D1(0-31),
              D2(0-31),
              D3(0-31),
              D4(0-31),
              D5(0-31),
              D6(0-31),
              D7(0-31),
                                    $address registers
              A0(0-31),
              A1(0-31),
              A2(0-31),
              A3(0-31),
              A4(0-31),
              A5(0-31),
              A6(0-31).
                                     $user stack pointer
              UA7(0-31),
                                     $supervisor stack
               SA7(0-31).
                                    pointer
               PC(0-31),
                                     $program counter
               SR(0-15),
                                     $status register
$ Internal registers (defined by authors)
                                    $prefetch register
              PFR(0-15),
Register,
                                    $instruction register
              IR(0-15),
                                    $function code register
              FC(0-2)
                                    $external data bus
              EXDBUF(0-15),
                                    buffer register
                                    Sexternal address bus
              EXABUF(0-15),
                                    buffer register
                                    $ALU buffer 1
              ALUBUF1(0-31),
                                    $ALU buffer 2
              ALUBUF2(0-31),
                                    $temporary storage for
              DTEMP(0-15),
                                    data
              DISREG(0-31),
                                     $temporary storage for
                                    displacement
                                     $temporary storage for
              SRTEMP(0-15),
                                     status register (used
                                     for exception
                                     processing)
                                     $temporary storage for
              IRTEMP(0-15),
                                     instruction register
                                     (used for exception
                                     processing)
                                     $temporary storage for
              TEMPADR(0-31).
                                     current cycle address
                                     (used for exception
                                     processing)
```

は、異ないのないのと、これはないないので、一つないのでは、一つないのでは、

(,

	ACTYPE(0-15),	<pre>\$temporary storage for access type information (used for exception processing)</pre>
	VECADR(0-23),	\$temporary storage for vector address (used for exception processing)
·	HANADR(0-31),	\$temporary storage for address of exception handler routine (used for exception processing)
	T(0-7),	\$control register, clock cycle counter (reset at end of each instruction)
	RESET, HALT,	<pre>\$reset flip-flop \$halt flip-flop</pre>
	RW,	\$read/write flip-flop
	ADÉNABLE,	\$address bus buffer enable, high impedance when low, enabled when high
	DBENABLE,	\$data bus buffer enable, high impedance when low, enabled when high
	ASN,	<pre>\$address strobe flip-flop</pre>
	LDSN,	\$lower data strobe
	UDSN,	<pre>flip-flop \$upper data strobe flip-flop</pre>
	DTACKN,	<pre>\$data transfer acknowledge flip-flop (from peripheral device)</pre>
	COUT,	\$carry flip-flop
	EXCEPT,	<pre>\$exception processing flip-flop</pre>
	READY	\$READY flip-flop (indicates processor is ready after power up or reset)
<pre>\$ Subregister Subregisters,</pre>	<pre>declarations PC(ADDR)=PC(0-23),</pre>	\$address field of program counter
	SR(TRACE)=SR(15), SR(MODE)=SR(13), SR(CARRY)=SR(0), SR(OVER)=SR(1), SR(ZERO)=SR(2),	<pre>\$trace bit \$mode selection bit \$carry bit \$overflow bit \$zero bit</pre>
	SR(NEG)=SR(3), SR(EX)=SR(4),	<pre>\$negative bit \$extend bit</pre>

```
SR(MASK)=SR(8-10),
              FC(SPACE)=FC(0-1),
                                    $address space of memory
                                    access
              FC(MODE)=FC(2),
                                    $user or supervisor mode
                                    bit
              PC(LOW)=PC(0-15),
                                    $low word of PC
              PC(HI)=PC(16-31).
                                    $high word of PC
              DO(LWORD)=DO(0-15),
                                    $low word of DO
                                    $low word of D1
              D1(LWORD)=D1(0-15),
              D2(LWORD)=D2(0-15),
                                    $low word of D2
              D3(LWORD) = D3(0-15),
                                    $low word of D3
              D4(LWORD) = D4(0-15),
                                    $low word of D4
              D5(LWORD)=D5(0-15),
                                    $low word of D5
              D6(LWORD) = D6(0-15),
                                    $low word of D6
              D7(LWORD)=D7(0-15),
                                    $low word of D7
              DISREG(HWORD)=
                                    $high word of DISREG
                  DISREG(16-31)
              DISREG(LWORD)=
                                    $low word of DISREG
              HANADR(LOW)=
                                    $1ow word of HANADR
                  HANADR(0-15),
              HANADR(HI)=
                                    $high word of HANADR
                  HANADR(16-31),
              TEMPADR(LOW)=
                                   $low word of TEMPADR
                  TEMPADR(0-15),
                                    $high word of TEMPADR
              TEMPADR(HI)=
                  TEMPADR(16-31)
$ Bus declarations
              IABUS(0-31),
                                    $internal address bus
Bus,
              IDBUS(0-31),
                                    $internal data bus
              DBUS(0-15),
                                    Sexternal data bus
              ABUS(0-23)
                                    Sexternal address bus
$ Decoder declarations
            A(0-3)=IR(14-15),
Decoders,
                                    $decoders used to decode
                                    various fields of the
                                    instruction register
              B(0-3)=IR(12-13),
              C(0-7)=IR(9-11),
              D(0-7)=IR(6-8),
              E(0-7)=IR(3-5),
              F(0-7)=IR(0-2)
              G(0-15)=IR(8-11),
              H(0-3)=IR(6-7)
                                    $control register (clock
              K(0-256)=T(0-7)
                                    cycle counter) decoder
$ Switch, memory, and clock declarations
                                    $power switch
Switch,
            POWER(ON,OFF)
Memory,
              M()=M(0-3388608,0-15)
Clock,
              P(1-2)
                                   $two phase clock
```

\$interrupt mask

Appendix C: ISP' Models of MC68000 Instructions

The ISP' descriptions for each of the modeled MC68000 instructions or exception sequences appear in this appendix. They are:

ISP' Model	Page
1. MOVE.W D1,D2	C-3
2. MOVE.W D1, (A1)	C-16
3. MOVE.L Dl,Al	C-31
4. MOVE.W Dl,(Al)+	C-44
5. MOVE.W D1,04(A1)	C-62
6. MOVE.W D1,04(A1,D7)	C-79
7. MOVE.W D1,\$2004	C-97
8. MOVE.W Al,D3	C-114
9. MOVE.W (A1),D2	C-128
10. MOVE.W (A1)+,D6	C-143
11. MOVE.W -(A1),D4	C-161
12. MOVE.W 04(A1),D1	C-179
13. MOVE.W 04(Al,D7),D2	C-196
14. MOVE.W \$2004,D5	C-214
15. MOVE.W \$2004,\$2008	C-231
16. MOVE.W #\$5555,D1	C-256
17. ADD.W D3,D5	C-271
18. BEQ START	C-291
19. BTST D1, (A1)	C-310
20 TLLECAL INCOMPLEMENT PROPERTON	0-210

21. ILLEGAL ADDRESS EXCEPTION

C-358

```
/¥
                                                */
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W D1,D2 INSTRUCTION
                                                */
/*
                                                */
/×
                                                */
/x
              Structure Declarations
                                                */
/x
                                                */
state
*/
/*
                                                */
           M68000 Programming Registers
/*
                                                */
DE0:73<31:0>,
                   ! 8 Data Registers
AE0:63<31:0>,
                   ! 7 Address Registers
                  ! User Stack Pointer
UA7<31:0>,
SA7<31:0>.
                   ! System Stack Pointer
PC<31:0>.
                  ! Program Counter
SR(15:0),
                   ! Status Register
/*
                                                */
/*
                                                */
           Temporary Internal Registers
14
                                                */
PFR<15:00.
                   ! Prefetch Register
IR<15:0>,
                   ! Instruction Register
                   ! Function Code Register
FC<2:0>.
EXDBUF(15:0),
                   ! External Data Bus Buffer Register
EXABUF<23:1>,
                   ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,
                   ! ALU Buffer 1
                   ! ALU Buffer 2
ALUBUF2<31:0>,
DTEMP<15:0>.
                   ! Temporary Data Storage
DISREG<31:0>,
                   ! Temporary Displacement Storage
SRTEMP<15:00.
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
                   ! Temporary Instruction Register Storage
IRTEMP(15:0).
                   ! (Exception Processing)
                   ! Temporary Cycle Address Storage
TEMPADR (31:0),
                   ! (Exception Processing)
ACTYPE<15:0>,
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>,
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:00.
                        1 Temporary Address Storage For
                        ! Exception Handler Routine
T<7:00,
                        ! Clock Cycle Counter
RESET,
                      ! Reset Flip-Flop
HALT.
                      ! Halt Flip-Flop
F:W.
                      ! Read/Write Flip-Flop
ADENABLE.
                      ! Address Bus Buffer Enable
DIBENABLE,
                      ! Nata Bus Buffer Enable
ASN,
                      ! Address Strobe Flip-Flop
LIISN.
                      ! Lower Data Strobe Flip-Flop
UIISN,
                      ! Upper Data Strobe Flip-Flop
LITACKN.
                      ! Nata Transfer Acknowledge Flip-Flop
COUT,
                      | Carry Flip-Flop
                      ! Exception Processing Flip-Flop
EXCEPT,
READY,
                      ! Ready Flip-Flop
/x
                                                              */
/*
                                                              */
       Model transformation modifications:
13
                                                              */
/×
                                                              */
           1) CDL decoder structure nonexistent in ISP' and un-
       necessary for model. Eliminated.
/*
                                                              ¥./
/*
           Multi-phase clock structure nonexistent in ISP'.
                                                              */
/*
       Operations on registers will provide its equivalent.
                                                              */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
/*
       register will provide its equivalent.
                                                              */
/*
           4) The declared bus structures are modeled with registers */
       without loss of model accurracy. This done to maintain model
                                                              */
/ X
/×
       equivalency and simplicity.
                                                              */
/*
           5) The memory word length was reduced from 16 to 8 bit
                                                              */
       words to coincide with the ECR's 32-Kbyte memory, to agree with*/
/×
/×
       their PC incrementation, and to enable the use of existing
                                                              */
       MC68000 assembler and linker/loader models. The memory was
/×
                                                              */
       also reduced from 8 Mwords to 32 Kbytes.
14
                                                              */
/*
                                                              */
1ABUS(31:0),
                        ! Internal Address Rus
IDBUS<31:00,
                        ! Internal Data Bus
                      ! Power Switch
SWITCH.
                      / Phase 1 Of Two-Phase Clock
FHI1,
FHI2;
                      ! Phase 2 Of Two-Phase Clock
port
/×
                                                              */
/×
                                                              */
              External Address and Data Rus
14
                                                              */
DBUS (15:0),
                        ! External Data Bus
ABUS<23:12;
                        ! External Address Bus(changed)
```

```
format
11
                                                            */
/*
                 Register Subfields
                                                            */
/*
                                                            */
PCADDR
          = FC<23:0/,
                        / Program Counter Address Field
ORTRACE
          = SR(15),
                        ! Trace Bit
          = SR<13>,
SEMODE
                        ! Made Selection Rit
SROARRY
          = SR(0),
                        ! Carry Bit
SROVER
                        ! Overflow Bit
          = SR<1>,
          = SR<2>,
SRZERO
                        ! Zero Bit
SENEG
          = SR<3>.
                         Negative Bit
SREX
          = SR<4>,
                        ! Extend Bit
SKMASK
          = SEK(10:8),
                        ! Interrupt Mask
FICSPIACE
                        ! Memory Access Address Space
          = FC<1:0>,
          = FC<2>,
FCMODE
                        ! User/Supervisor Mode Bit
F'CLOW
          = PC(15:0),
                        ! FC Low Word
F:CHI
                        ! PC High Word
          = PC<31:16>,
DOLWORD
          = DE03<15:0>,
                        ! DEO3 Low Word
DILWORD
          = DE13<15:0>.
                        ! D[1] Low Word
DOLWORD
          = 10[2](15:0),
                         D[2] Low Word
DBLWORD
          = DE33<15:0>,
                        ! D[3] Low Word
DALWORD
          = D[4] < 15:0>
                        ! II[4] Low Word
DSLWORD
          = DESJ<15:0>.
                        ! DISD Low Word
DISCHORD
          = DE63<15:0>,
                        ! DI63 Low Word
DZLWORD
                        1 III73 Low Word
          = DE73<15:0>,
DISREGHWORD = DISREG (31:16>,! DISREG High Word
DISREGLWORD = DISREG(15:0), ! DISREG Low Word
HANALIF.LOW
          = HANADR<15:0>, ! HANADR Low Word
HANADEHI
          = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADENT
          = TEMPADR (31:16); ! TEMPADR High Word
Memory
1*
                                                            */
/*
                  16k 16-kit Word Internal Memory
                                                            */
                                                            */
ME0:327673<7:0>;
macro
14
                                                            */
/¥
                 Logic Level Macros
                                                            */
13
                                                            */
```

```
= 0 %,
    = 1 %,
ti 1
oft
    = 0 %,
on
    = 1 %,
clear = 0 %;
/*
  Fower On and Initialization. This process was not modeled but is
                                                         */
  added to initialize signals and registers.
                                                        */
/*
                                                         */
power_on_initialize :=
      SWITCH = on;
                                ! Turn Power On
      next;
                                 ! Execute Assignment
      REALTY = lo;
                                 ! System Not Ready
      RESET = lo:
                                ! Assert Reset For
      delay(100);
                                 ! 100 Miliseconds(Active Low)
      RESET = hi;
                                 ! Deactivate Reset
      next;
                                 ! Execute Pending Assignments
      ASN = hi;
                                ! Initialize Address Strobe
      LIBN = hi;
                                ! Initialize Lower Data Strobe
      UDSN = hi:
                                ! Initialize Upper Data Strobe
      D'FACKN = hi:
                                 ! Initialize Data Transfer Acknowledge
      RW = hi:
                                ! Initialize Read/Write(Read On High)
      DBUS = Oxffff;
                                ! Place Data Bus In High Impedance State
      M[4110] = 0 \times ff;
                                ! Place Memory Locations Following The
      ME41113 = 0 \times f;
                                 ! JMP Instruction In A High State
      HALT = hi;
                                 ! Initialize Halt Flip-Flop(Active
                                 ! Low)
      T = 0;
                                 ! Initialize Clock Cycle Counter
      READY = ti:
                                ! System Ready
      /*
                                                         */
      /*
            Routine Initialization Per Hamby and Guillory
                                                         */
      14
                                                         */
      SEMODE = 16;
                                ! Set Status Register To User Mode
      D[1] = 0::55555555;
                                ! Place Hex 5555555 Into D[1]
      10001 = 0x1000;
                                ! Place Hex 1000 Into ACO]
      PC = 0x1000;
                                ! Place Hex 1000 Into Program Counter
      next
                                ! Execute Assignments
      )
/*
  Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                        */
   to retrieve modeled instructions for simulation and analysis. It
/*
                                                        */
   was fashsioned from the Read Cycle described by Hamby and Guillory */
```

```
*/
/* on page VI-15 of their thesis.
/*
fetch_initial_instruction :=
     ! Phase 1 Of
    PHI1 = hi:
    PH12 = 16;
                                    ! Clock Cycle 0
    KW = hi;
                                    ! Memory Read
     ANENAM E = lo;
                                     ! Disable Address Bus Buffer
     DEENABLE = 10;
                                    ! Disable Data Bus Buffer
                                    ! Place PC On Internal Address
     TABUS = FC;
                                    ! Execute Fending Assignments
    next;
                                    ! Phase 2 Of
    PHI1 = lo;
                                    ! Clock Cycle 0
    FHI2 = hi;
                                    ! Enable Address Bus Buffer
     ADENABLE = hi;
    EXABUF = labus;
                                    ! Gate Internal Address Rus
                                    ! Into External Address Buffer
                                    ! User Mode
    FCMODE = SRMODE;
                                    ! Accessing Program
     FCSFACE = 2;
                                    ! Execute Impending Assignments
    next;
     ABUS = EXABUF;
                                    ! Address Placed On Bus(Added)
                                    ! Execute Pending Assignments
     next;
     T = 1:
                                    ! Clock Cycle 1
                                     ! Execute Assignment
     next;
                                    ! Phase 1 Of
     FHII1 = hi;
     PHI2 = 10;
                                    ! Clock Cycle 1
                                    ! Assert Address Strobe
     ASN = 10:
                                    ! Assert Lower Data Strobe
     LISN = lo;
                                    ! Assert Upper Data Strobe
     UIISN = lo;
     DRENABLE = hi;
                                    ! Enable Data Rus
                                    Execute Pending Assignments
     next;
     IH11 = 10;
                                    ! Phase 2
     PHI2 = hi;
                                    ! Of Clock Cycle 1
     next;
                                     ! Execute Pending Assignments
     T = 2;
                                     ! Clock Cycle 2
                                     ! Execute Assignment
     next;
     PHI1 = hi;
                                     ! Phase 1
     PHI2 = 10;
                                     ! Of Clock Cycle 2
                                     ! Wast for Memory To Place
     while DTACKN edl hi
                                     1 Data On The Rus
```

```
next;
                               ! Execute Impending Assignments
     PH11 = 10:
                               ! Phase 2
     PHI2 = hi;
                               1 Of Clock Cycle 2
                               ! Execute Assignments
     next;
     ! Clock Cycle 3
     next;
                               ! Execute Assignment
     PH11 = hi;
                               ! Phase 1
     PHI2 = 1o;
                               ! Of Clock Cycle 3
     DBUS(15:8> = MEABUSD;
                               ! Memory Places Instruction
     IGNUS <7:0> = MEARUS + 13;
                               ! On Data Bus And
     DYACKN = lo;
                               ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
     next;
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
     next;
                               ! Execute Impending Assignments
T = 3;
                                ! Clock Cycle 3
next;
                                ! Execute Assignment
                               ! Phase 2
PH11 = 10;
                               ! Of Clock Cycle 3
PHI2 = ni;
EXEBUF = DBUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
T = 4;
                               ! Clock Cycle 4
next:
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                                ! Of Clock Cycle 4
PHI2 = 10;
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Flaced
                               In Prefetch Register
                                ! Execute Fending Assignments
next;
PHI1 = lo:
                               ! Phase 2
                                ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                                ! Deactivate Address Strobe
LDSN = hi;
                                ! Neactivate Lower Nata Strobe
                                ! Deactivate Upper Data Strobe
UDSN = hi;
IR = PFR;
                                ! Contents Of Prefetch Register
                                ! Are Flaced Into Instruction
                                ! Register
```

```
! Deactivate Data Transfer(Added)
     DITACKN = hi;
                                         ! Acknowledge
     PC = PC + 2;
                                         ! Increment Program Counter
     next;
                                         ! Execute Fending Assignments
     T = 0
                                         * Reset Clock Cycle Counter
     )
                                         ! MOVE.W D1,D2
MOVE :=
     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10:
                                         ! Clock Cycle 0
     DBUS = Oxffff;
                                         ! Place Data Bus In High Impedance
                                         ! Memory Read
     F:W = hi;
     ADENABLE = 10;
                                         ! Disable Address Bus Buffer
     DRENABLE = 10;
                                         ! Disable Data Bus Buffer
                                         ! Place PC On Internal Address
     IABUS = FC;
                                         Rus
                                         ! Flace Low Word From D[1] Onto
     IDBUS = DILWORD;
                                         ! Internal Data Bus
     next;
                                         ! Execute Pending Assignments
                                         ! Phase 2 Of
     FHI1 = lo;
     PHI2 = hi;
                                         ! Clock Cycle 0
     ADENABLE = hi;
                                         ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
     FCMODE = SRMODE;
                                         ! User Mode
     FCSFACE = 2:
                                         ! Accessing Program
     SRCARRY = 10:
                                         ! Clear Status Register Carry Bit
                                         ! Clear Status Register Overflow Bit
     SROVER = lo;
     SRZERO = lo;
                                         ! Clear Status Register Zero Bit
     SRNEG = 10;
                                         ! Clear Status Register Negative Bit
     D2LWORD = IDBUS;
                                         ! Place Data From Internal Data Bus
                                         ! Into Low Word Of DC23
                                         ! Execute Impending Assignments
     next:
     ARUS = EXARUF;
                                         ! Address Flaced On Bus(Added)
     next;
                                         ! Execute Pending Assignments
     T = 1;
                                         ! Clock Cycle 1
     next;
                                         ! Execute Assignment
                                         ! Phase 1 Of
     PH11 = h1;
                                         ! Clock Cycle 1
     FHI2 = 10;
                                         ! Assert Address Strobe
     ASN = 10;
     LIISN = lo:
                                         ! Assert Lower Data Strobe
                                         ! Assert Upper Data Strobe
     UDSN = 10;
     DBENABLE = hi;
                                         ! Enable Data Bus
```

```
if D2LWORD eql 0
                              ! Set Status Register Zero Bit
  SRZERO = hi;
                              ! If Moved Data Is Zero
                              ! Execute Pending Assignments
next;
PHI1 = lo;
                              ! Phase 2
PHI2 = hi:
                              ! Of Clock Cycle 1
if DE23<15>
                              ! Set Status Register Negative
  SRNEG = hi;
                              ! Bit If Moved Data Is Negative
next:
                              ! Execute Pending Assignments
T = 2;
                              ! Clock Cycle 2
next:
                              ! Execute Assignment
FHI1 = h1;
                              ! Phase 1
FHI2 = 10;
                              ! Of Clock Cycle 2
while DTACKN eql hi
                              ! Wait For Memory To Flace
    (
                              ! Data On The Bus
                              ! Execute Impending Assignments
    next;
                              ! Phase 2
    FHI1 = lo;
                              ! Of Clock Cycle 2
    PHI2 = hi;
                              ! Execute Assignments
    next;
    T = 3;
                              ! Clock Cycle 3
                              ! Execute Assignment
    next;
    PH11 = h1;
                              ! Phase 1
    PHI2 = lo;
                             ! Of Clock Cycle 3
    DBUS<15:8> = MCARUSD;
                              ! Memory Flaces Instruction
    DBUS<7:0> = MEABUS + 1];
                              ! On Data Bus And
                              ! Asserts DTACKN(Added)
    DITACKN = 10;
                              ! Execute Pending Assignments
    next;
    T = 2
                              ! Return To Phase 2
                              ! Of Clock Cycle 2
    );
    next;
                              ! Execute Impending Assignments
T = 3;
                              ! Clock Cycle 3
next:
                              ! Execute Assignment
F'HI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 3
PH12 = hi;
EXDRUF = DRUS;
                              ! Instruction On Data Bus
                              ! Is Placed In External Data
                              ! kus Ruffer
                              ! Execute Pending Assignments
next;
```

```
T = 4;
                                           ! Clock Cycle 4
     next;
                                           ! Execute Assignment
                                           ! Phase 1
     FHI1 = hi;
     FH12 = 10;
                                           ! Of Clock Cycle 4
     PER = EXDBUF;
                                           ! The Contents Of The External
                                           ! Data Bus Buffer Are Flaced
                                           ! In Prefetch Register
                                           / Execute Fending Assignments
     next;
     FHI1 = lo;
                                           ! Phase 2
     PHI2 = 51;
                                           ! Of Clock Cycle 4
                                           ! Deactivate Address Strobe
     ASN = hi;
     LUSN = hi;
                                           ! Neactivate Lower Nata Strobe
     UDSN = hi;
                                           ! Deactivate Upper Data Strobe
     IR = FFE;
                                           ! Contents Of Prefetch Register
                                           ! Are Placed Into Instruction
                                           ! Register
                                           ! Deactivate Data Transfer(Added)
     DTACKN = hi;
                                           ! Acknowledge
     PC = PC + 2;
                                           ! Increment Program Counter
     next;
                                           ! Execute Impending Assignments
     T = 0
                                           ! Reset Clock Cycle Counter
                                           ! JMF (AO)
.jmp :=
      PHI1 = hi;
                                           ! Phase 1 Of
     FHI2 = 10;
                                           ! Clock Cycle 0
     DBUS = Oxffff;
                                           ! Place Data Bus In A High Impedance
     IW = hi;
                                           ! Memory Read
     ADENABLE = lo;
                                           ! Disable Address Bus Buffer
     DIMENABLE = 10;
                                           ! Nisable Nata Bus Buffer
     IABUS = PC;
                                           ! Place PC On Internal Address
                                           ! Bus
     next;
                                           ! Execute Pending Assignments
     f'HI1 = lo;
                                           ! Phase 2 Of
     PHI2 = hi;
                                           ! Clock Cycle O
     ADENABLE = hi;
                                           ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                           ! Gate Internal Address Bus
                                           ! Into External Address Buffer
     FCMODE = SRMODE;
                                           ! User Mode
     FICSPACE = 2;
                                           ! Accessing Program
     next;
                                           ! Execute Pending Assignments
     ABUS = EXABUF;
                                           ! Address Flaced On Rus(Added)
     next;
                                           ! Execute Pending Assignments
```

```
T = 1;
                              ! Clock Cycle 1
                              ! Execute Assignment
next;
                             ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                              ! Clock Cycle 1
ASN = 10;
                              ! Assert Address Strobe
LDSN = 10;
                              ! Assert Lower Data Strobe
                              ! Assert Upper Data Strobe
UDSN = 10;
                             ! Move Jump Address From A[0]
TABUS = ALOJ;
                              ! To Internal Address Buffer
                              ! Enable Data Rus
DBENABLE = hi;
next;
                             ! Execute Pending Assignments
PHI1 = lo;
                             ! Phase 2
FHI2 = hi;
                             ! Of Clock Cycle 1
PC = IABUS;
                             ! Place Jump Address Into Program
                              ! Counter
next;
T = 2;
                              ! Clock Cycle 2
                              ! Execute Assignment
next;
FH11 = h1;
                             ! Phase 1
                              ! Of Clock Cycle 2
PHI2 = 10;
while DTACKN eql hi
                              ! Wait For Memory To Place
                              ! Nata On The Bus
    (
    next;
                             ! Execute Impending Assignments
    PHI1 = lo;
                             ! Phase 2
    PHI2 = bi;
                             ! Of Clock Cycle 2
    next;
                              ! Execute Assignments
    T = 3:
                              ! Clock Cycle 3
    next;
                              ! Execute Assignment
    FHI1 = hi:
                              ! Phase 1
    PHI2 = lo;
                              ! Of Clock Cycle 3
    DBUS<15:8> = MEABUS];
                              ! Memory Places Instruction
    DBUS (7:0> = MEABUS + 13;
                             ! On Data Bus And
    DITACKN = 10;
                              ! Asserts DTACKN(Added)
    next; .
                              ! Execute Pending Assignments
    T = 2
                             ! Return To Phase 2
                              ! Of Clock Cycle 2
    );
                              ! Execute Impending Assignments
    next:
T = 3:
                             ! Clock Cycle 3
```

```
! Execute Assignment
next;
PHI1 = 16;
                                   ! Phase 2
                                   ! Of Clock Cycle 3
PHI2 = hi:
                                   ! Instruction On Data Rus
EXDRUF = DEUS:
                                   ! Is Placed In External Data
                                   ! Rus Buffer
                                   ! Execute Pending Assignments
nest;
! Clock Cycle 4
T = 4;
                                   ! Execute Assignment
next;
PHI1 = hi;
                                   ! Phase 1
                                   ! Of Clock Cycle 4
PHI2 = lo;
next:
                                   ! The Contents Of The External
PER = EXBBUE;
                                   ! Data Rus Buffer Are Placed
                                   ! In Prefetch Register
next;
                                   ! Execute Pending Assignments
                                   ! Phase 2
FHI1 = lo;
                                   1 Of Clock Cycle 4
PHI2 = hi;
                                   ! Neactivate Address Strobe
ASN = hi;
                                   ! Deactivate Lower Data Strobe
LDSN = hi;
UNSN = hi;
                                   ! Deactivate Upper Data Strobe
                                   ! Deactivate Data Transfer
DIACKN = hi;
                                   ! Acknowledge(Added)
next;
T = 5:
                                   ! Clock Cycle 5
next;
                                   ! Execute Previous Assignment
                                   ! Phase 1 Of
PH11 = hi;
                                   ! Clock Cycle 5
PHI2 = 16;
                                   ! Memory Read
£₩ = h1;
                                   ! Disable Address Bus Buffer
ADENABLE = 10;
                                   ! Disable Data Bus Ruffer
DEENABLE = 10;
TABUS = PC;
                                   ! Flace FC On Internal Address
                                   1 Bus
next;
                                   ! Execute Pending Assignments
Fill1 = lo;
                                   ! Phase 2 Of
                                   ! Clock Cycle 5
PHI2 = hi;
ADENABLE = hi:
                                   ! Enable Address Bus Buffer
[CMUDIL = SEMUDIE;
                                   ! User Mode
FOSPACE = 2;
                                   ! Accessing Program
                                   ! Gate Internal Address Bus
EXABUF = TABUS;
ne it;
                                   Into External Address Buffer
ABUS = EXABUF;
                                   ! Address Placed On Bus(Added)
nert;
                                   ! Execute Pending Assignments
```

```
T = 6;
                               / Clock Cycle 6
next;
                               ! Execute Assignment
                               ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                               ! Clock Cycle 6
ASN = lo;
                               ! Assert Address Strobe
LIISN = 10;
                               ! Assert Lower Data Strobe
UDSN = 10;
                               i Assert Upper Data Strobe
DBENABLE = hi:
                               ! Enable Data Bus
                               ! Execute Pending Assignments
next:
                               ! Phase 2
FHII = lo;
PHI2 = hi;
                               1 Of Clock Cycle 6
next;
                               ! Execute Pending Assignments
T = 7;
                               ! Clock Cycle 7
                               ! Execute Assignment
next;
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 7
PHI2 = 10:
while NTACKW eql hi
                               ! Wait For Memory To Flace
                               ! Data On The Bus
     (
                               ! Execute Impending Assignments
    next;
     PHI1 = 10;
                               1 Phase 2
                               ! Of Clock Cycle 7
    PHI2 = ni;
     next;
                               ! Execute Assignments
     T = 8:
                               ! Clock Cycle 8
                               ! Execute Assignment
     next;
                               ! Phase 1
     PHI1 = hi;
     FHI2 = 10;
                               ! Of Clock Cycle 8
     DBUS(15:8> = MCABUS);
                               ! Memory Places Instruction
    168US (7:0) = MEABUS + 13;
                               ! On Data Rus And
                               ! Asserts DTACKN(Added)
     DITACKN = 10;
     next;
                               ! Execute Pending Assignments
     7 = 7
                               ! Return To Phase 2
                               ! Of Clock Cycle 7
     1;
     next;
                               ! Execute Impending Assignments
7 = 8;
                               ! Clock Cycle 8
next;
                               ! Execute Assignment
                               ! Phase 2
f'HI1 = lo;
FHI2 = hi;
                               ! Of Clock Cycle 8
EXERUF = DBUS;
                               ! Instruction On Data Rus
```

```
! Rus Buffer
                                          ! Execute Pending Assignments
     next;
     T = 9;
                                          ! Clock Cycle 9
                                          ! Execute Assignment
     next;
     PHI1 = hi:
                                          ! Phase 1
     FHI2 = 10;
                                          ! Of Clock Cycle 9
                                          ! The Contents Of The External
     PFR = EXDBUF;
                                          ! Nata Bus Buffer Are Placed
                                          ! In Frefetch Register
     next;
                                          ! Execute Pending Assignments
                                          ! Phase 2
     PHI1 = 10;
     FHI2 = hi;
                                          ! Of Clock Cycle 9
     AGN = hi;
                                          ! Deactivate Address Strobe
                                          ! Deactivate Lower Data Strobe
     LIISN = hi;
                                          ! Deactivate Upper Data Strobe
     UDSN = hi;
                                          ! Increment Program Counter
     FC = FC + 2;
                                          ! Place Contents Of Prefetch
     IR = PFR:
                                          ! Register Into Instruction
                                          ! Register
     DITACKN = hi;
                                          ! Deactivate Data Transfer
                                          ! Acknowledge(Added)
                                          ! Execute Pending Assignments
     next;
     \Upsilon = 0
                                          ! Reset Clock Cycle Counter
decode_execute_profetch :=
                       case IR
                           032001: move
                                          ! MOVE.W D1,D2
                           047320: јыр
                                          ! JMP (AO) If IR = Octal Value
                       esac
main :=
    power_on_initialize;
    fetch initial instruction;
    while READY eql hi
          decode_execute_prefetch
```

! Is Placed In External Data

```
/*
                                                    */
/×
     MOTOROLA MC68000 MODEL OF THE MOVE.W D1, (A1) INSTRUCTION
                                                    */
/*
                                                    */
1%
                                                    */
/*
                                                    */
               Structure Declarations
/×
                                                     */
state
*/
/*
/*
                                                    */
            m68000 Programming Registers
/*
                                                    */
ACO:73<31:0>,
                    ! 8 Data Registers
A[0:63<31:0>,
                    ! 7 Address Registers
UA7<31:0>,
                    ! User Stack Pointer
SA7<31:0>,
                    ! System Stack Pointer
PC<31:0>,
                    ! Program Counter
SR<15:0>.
                    ! Status Register
1*
                                                    */
                                                    */
/*
            Temporary Internal Registers
/*
PFR<15:0>,
                     ! Frefetch Register
IR (15:0),
                    ! Instruction Register
                     ! Function Code Register
FC<2:0>,
                    ! External Data Bus Buffer Register
EXDBUF(15:0),
EXABUTE 23:1>.
                     ! External Address Bus Buffer Register(changed)
                    ! ALU Buffer 1
ALUBUF1<31:0>,
nLURUF2 (31:0),
                    □ ALU Buffer 2
INTEMPORTS:0.,
                     ! Temporary Data Storage
                     ! Temporary Displacement Storage
1:15REG<31:0>,
SRTEMP<15:00.
                     ! Temporary Status Register Storage
                     ! (Exception Processing)
                     ! Temporary Instruction Register Storage
IRTIME 15:00.
                     ! (Exception Processing)
TEMPADR (31:0),
                     ! Temporary Cycle Address Storage
                     ! (Exception Processing)
ACTYPE(15:0),
                     ! Temporary Access Type Storage
                     ! (Exception Processing)
VECADR<23:0>,
                     ! Temporary Vector Address Storage
                     ! (Exception Processing)
```

```
! Temporary Address Storage For
HANADR (31:0),
                        ! Exception Handler Foutine
                        ! Clock Cycle Counter
T<7:0>.
RESET.
                      ! Reset Flip-Flop
HALT,
                      ! Halt Flip-Flop
RW.
                      1 Read/Write Flip-Flop
ADENABLE,
                      ! Address Bus Buffer Enable
                      ! Nata Bus Buffer Enable
DIBENABLE,
ASN.
                      ! Address Strobe Flip-Flop
LIISN.
                      ! Lower Nata Strobe Flip-Flop
UUSN,
                      ! Upper Data Strobe Flip-Flop
                     ! Data Transfer Acknowledge Flip-Flop
DITACKN.
COUT,
                      ! Carry Flip-Flop
EXCEPT,
                      ! Exception Processing Flip-Flop
READY,
                      ! Ready Flip-Flop
/x
                                                              */
/*
      Model transformation modifications:
                                                              */
/*
                                                              */
14
           1) CBL decoder structure nonexistent in ISP' and un-
                                                              */
/×
      necessary for model. Eliminated.
                                                              x /
/×

    Multi-phase clock structure nonexistent in ISP'.

/*
      Operations on registers will provide its equivalent.
           3) Switch structure nonexistent in ISP'. Operation on a
/*
                                                              */
1*
                                                              x /
       register will provide its equivalent.
/ ¥

 The declared bus structures are modeled with registers */

       without loss of model accurracy. This done to maintain model
14
                                                              */
/*
      equivalency and simplicity.
                                                              */
1%
           5) The memory word length was reduced from 16 to 3 bit
                                                              x/
/×
      words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/¥
      their FC incrementation, and to enable the use of existing
                                                              */
11
      MC68000 assembler and linker/loader models. The memory was
                                                              14/
       also reduced from 8 Mwords to 32 Kbytes.
                                                              */
/*
/*
                                                              */
TABUS<31:00,
                        ! Internal Address Rus
                        ! Internal Data Bus
IDBUS<31:0>.
twast<7:0>,
                        + Wait Cycle Counter
SWITCH,
                      ! Power Switch
PHI1,
                      ! Phase 1 Of Two-Phase Clock
PH12;
                      ! Phase 2 Of Two-Phase Clock
port
*/
                                                              ¥/
/*
             External Address and Data Bus
1%
                                                              */
DBUS (15:0>,
                        - External Data Bus
```

```
ABUS<23:1>;
                     ! External Adaress Bus(changed)
format
14
                                                         #/
/*
                                                         */
                 Register Subfleids
/¥
                                                         */
PCADDR
         = PC<23:0>,
                       ! Program Counter Address Field
SKTRACE
         = SR<15>,
                       ! Trace Bit
         = SR<13>,
                       ! Mode Selection Bit
SEMODIE
                       ! Carry Bit
SRCARRY
         = SR<0>,
SROVER
         = SR<1>,
                      ! Overflow Bit
SRZERO
         = SR<2>,
                       ! Zero Bit
SENEG
         = SR<3>,
                       ! Negative Bit
SREX
                       ! Extend Bit
         = SR<4>,
SRMASK
         = SR<10:8>.
                      ! Interrupt Mask
FICSPIACE
         = FC(1:0),
                        Memory Access Address Space
FCMUDE
         = FC(2),
                      ! User/Supervisor Mode Bit
         = PC<15:00,
                       ! PC Low Word
FICLUW
F'CHI
         ≈ PC<31:16>,
                       ! PC High Word
HOLWORD
         = DE03(15:0/,
                       ! II[0] Low Word
DILWORD
         = D[1](15:0),
                      ! D[1] Low Word
DI2LWORD
         = BE23(15:0),
                       ! DECO Low Word
DBLWORD
         = DEST (15:0),
                      1 DE31 Low Word
I/4LWORD
         = I([4])(15:0),
                      1 DEAD Low Word
DSLWORD
         = DE53 (15:0),
                      1 DEST Low Word
I & LWORD
         = IL63 15:60,
                      ! Blél Low Word
         = DE73:15:00,
D7LWORD
                      ! DE73 Low Word
MISREGHWORD = DISREG (31:16), ! DISREG High Word
DISREGLWORD = DISREG(15:0), ! DISREG Low Word
         = HANADR (15:0>, ! HANADR Low Word
HANATIKLOW
HANGDRHI
         = HANADR 31:16 , HANADR High Word
TEMPADRLOW = TEMPADR<15:0>, ! TEMPADR Low Word
TEMPANCHI
         = TEMPADR<31:160;! TEMPAUK High Word
memory
/*
                                                         */
/x
                 16K 16-Bit Word Internal Memory
                                                         */
/*
                                                         */
mE0:32767347:07:
macho
/*
                                                         */
1*
                Logic Level Macros
                                                         */
```

```
10
    = 0 %,
    = 1 4,
ħi
off
    = 0 %.
    = 1 %,
CLD
clear = 0 %;
/* Power On and Initialization. This process was not modeled but is
                                                       */
                                                       x./
  added to initialize signals and registers.
                                                       */
/*
power_on_initialize :=
      SWITCH = on;
                                ! Turn Power On
      next;
                                ! Execute Assignment
                                ! System Not Ready
      READY = 10;
      RESET = lo;
                                ! Assert Reset For
                                ! 100 Miliseconds(Active Low)
      delay(100);
      RESET = hi:
                                ! Deactivate Reset
      next:
                                ! Execute Pending Assignments
                                ! Initialize Address Strobe
      ASN = hi;
                                ! Initialize Lower Data Strobe
      LISN = bi:
      UDSN = ha;
                                ! Initialize Upper Data Strobe
                                ! Initialize Data Transfer Acknowledge
      DIACKN = hi;
      RW = hi;
                                ! Initialize Read/Write(Read On High)
                                ! Flace Data Rus In High Impedance State
      DBUS = 0xffff;
      M[4110] = 0xff;
                                ! Place Memory Locations Following The
                                ! JMP Instruction In A High State
      ME41113 = 0xff;
      HALT = hi;
                                ! Initialize Halt Flip-Flop(Active
                                ! Low)
                                ! Initialize Clock Cycle Counter
      T = 0;
      READY = hi;
                                ! System Ready
      */
      ′¥
      /*
           Routine Initialization Per Hamby and Guillory
                                                       */
      SRMODE = 10;
                                ! Set Status Register To User Mode
      D[1] = 0.55555555;
                                ! Place Hex 5555555 Into D[1]
      AI03 = 0x1000:
                                ! Place Hex 1000 Into A[0]
                                ! Store Data At Hex 2000
      h[1] = 0.2000;
                                ! Place Hex 1000 Into Program Counter
      FC = 0 \times 1000;
                                ! Execute Assignments
      next
      )
Initial Fetch Cycle. This cycle was not modeled but is necessary
```

```
/* to retrieve modeled instructions for simulation and analysis. It
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/w on page VI-15 of their thesis.
                                                           */
/*
                                                           */
fetch_init;al_instruction :=
     ! Phase 1 Of
     PHI1 = hi;
     FHI2 = lo;
                                     ! Clock Cycle 0
     RW = hi;
                                     ! Memory Read
     ADENABLE = 10;
                                     ! Disable Address Bus Buffer
                                     ! Disable Data Bus Ruffer
     DRENABLE = 10:
                                     ! Place PC On Internal Address
     TABUS = PC;
                                     1 Kus
                                     ! Execute Pending Assignments
     next;
     FHI1 = lo;
                                     ! Phase 2 Of
                                     ! Clock Cycle 0
     PHI2 = hi;
                                     ! Enable Address Bus Buffer
     ADENABLE = hi;
                                     ! Gate Internal Address Bus
     EXABUF = IABUS;
                                     ! Into External Address Buffer
     FCMODE = SRMODE;
                                     ! User Mode
     FCSPACE = 2;
                                     ! Accessing Program
                                     ! Execute Impending Assignments
     next;
     ABUS = EXABUF;
                                     ! Address Placed On Rus(Added)
                                     ! Execute Pending Assignments
     T = 1;
                                     ! Clock Cycle 1
     next;
                                     ! Execute Assignment
                                     ! Phase 1 Of
     PHI1 = hi;
                                     ! Clock Cycle 1
     f'HI2 = lo;
     ASN = 10;
                                     ! Assert Address Strobe
     LUSN = lo;
                                     ! Assert Lower Data Strobe
     UDSN = 10;
                                     ! Assert Upper Data Strobe
                                     ! Enable Data Bus
     DBENABLE = hi;
                                     ! Execute Pending Assignments
     next;
     PHII = lo;
                                     ! Phase 2
     PHI2 = hi;
                                     ! Of Clock Cycle 1
                                     ! Execute Pending Assignments
     next:
     T = 2:
                                     ! Clock Cycle 2
     next;
                                     ! Execute Assignment
     PHI1 = ha;
                                     ! Phase 1
     PHI2 = 10;
                                     ! Of Clock Cycle 2
```

THE PROPERTY OF THE PROPERTY O

```
! Wait For Memory To Place
while DTACKN eql hi
                                ! Data On The Bus
    next;
                               ! Execute Impending Assignments
    PHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 2
    PHI2 = hi;
     next:
                               ! Execute Assignments
     T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
     next;
    PH11 = hi;
                               ! Phase 1
    PHI2 = 10;
                               ! Of Clock Cycle 3
                               ! Memory Flaces Instruction
     DBUS<15:8> = MEABUS3;
                               ! On Data Bus And
    DBUS<7:0> = MEABUS + 13;
    DTACKN = lo;
                               ! Asserts DTACKN(Added)
    next;
                               ! Execute Pending Assignments
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
     next;
                               ! Execute Impending Assignments
T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
next;
FH11 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 3 -
PHI2 = hi;
                                ! Instruction On Nata Bus
EXDBUF = DBUS;
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
! Clock Cycle 4
T = 4:
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 4
                               ! The Contents Of The External
PFR = EXDBUF;
                               ! Data Bus Buffer Are Placed
                                ! In Frefetch Register
                               ! Execute Pending Assignments
next:
PHI1 = 10:
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 4
ASN = ni;
                               ! Deactivate Address Strobe
LISN = hi;
                               ! Beactivate Lower Data Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
IR = PFR;
                                ! Contents Of Prefetch Register
```

```
! Are Placed Into Instruction
                                         ! Register
     DTACKN = hi;
                                         ! Deuctivate Dota Transfer(Added)
                                         ! Acknowledge
     PC = PC + 2;
                                        1 Increment Program Counter
     next;
                                        1 Execute Pending Assignments
     T = 0
                                        ! Reset Clock Cycle Counter
                                        ! MOVE.W 11, (A1)
move :=
     PHI1 = hi;
                                         ! Phase 1 Of
     FHI2 = 10;
                                        ! Clock Cycle 0
     DBUS = 0xffff;
                                        ! Place Data Bus In High Impedance
     RW = hi;
                                         ! memory Read
     ADENABLE = 10;
                                        ! Disable Address Bus Buffer
                                         ! Disable Data Bus Buffer
     DIBENABLE = 10;
                                        ! Place PC On Internal Address
     IABUS = PC;
     next;
                                        ! Execute Pending Assignments
                                        ! Phase 2 Of
     f'HI1 = lo;
     PH12 = h1;
                                        ! Clock Eycle O
                                        ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS:
                                        ! Gate Internal Address Rus
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSPACE = 2;
                                        ! Accessing Program
     next;
                                        ! Execute Impending Assignments
     ABUS = EXABUF;
                                        ! Address Flaced On Bus(Added)
                                        ! Execute Pending Assignments
     next;
     T = 1;
                                         ! Clock Cycle 1
     next;
                                         ! Execute Assignment
                                        ! Phase 1 Of
     PHI1 = hi;
                                         ! Clock Cycle 1
     PH12 = 10;
     ASN = 10;
                                         ! Assert Address Strobe
     LIGN = lo;
                                         ! Assert Lower Data Strobe
                                         ! Assert Upper Data Strobe
     UDSN = lo:
     DBENABLE = hi:
                                         ! Enable Nata Rus
     next;
                                         ! Execute Pending Assignments
     FHI1 = lo:
                                        ! Phase 2
     PHI2 = hi;
                                        ! Of Clock Cycle 1
     next;
                                        ! Execute Pending Assignments
```

```
T = 2;
                              ! Clock Cycle 2
next;
                              ! Execute Assignment
PHII = ha;
                             ! Phase 1
                             ! Of Clock Cycle 2
fHI2 = 10;
                             ! Wait For Memory To Place
while DTACKN eql hi
                             ! Data On The Bus
    next:
                             ! Execute Impending Assignments
    PHI1 = lo;
                             ! Phase 2
                             ! Of Clock Cycle 2
    PHI2 = hi;
                             1 Execute Assignments
    next;
    \tilde{1} = 3;
                              ! Clock Cycle 3
    next;
                             ! Execute Assignment
    PHII = hi;
                             ! Phase 1
    PHI2 = 10;
                              ! Of Clock Cycle 3
                             ! Memory Places Instruction
    DBUS<15:8> = MEABUS];
    DBUS(7:0) = MEABUS + 11;
                             ! On Data Rus And
    DIACKN = 16;
                              ! Asserts DTACKN(Added)
                              ! Execute Pending Assignments
    next:
    T = 2
                              ! Return To Phase 2
                             ! Of Clock Cycle 2
    );
                             ! Execute Impending Assignments
    next;
! Clock Cycle 3
i = 3;
                              ! Execute Assignment
next;
FHII = 10;
                             ! Phase 2
PHT2 = h1;
                             ! Of Clock Cycle 3
EXDBUF = DBUS;
                              ! Instruction On Data Bus
                             ! Is Placed In External Data
                              ! Rus Ruffer
                              ! Execute Pending Assignments
next;
1 = 4;
                              ! Clock Cycle 4
                              ! Execute Assignment
next;
                              ! Phase 1
PHI1 = hi;
PHI2 = 10;
                              ! Of Clock Cycle 4
                             ! The Contents Of The External
PFR = EXDBUF;
                             ! Nata Bus Buffer Are Flaced
                              ! In Prefetch Register
                              ! Execute Pending Assignments
next;
```

```
! Phase 2
PHI1 = 10;
                                    ! Of Clock Cycle 4
PHIC = hi;
                                    ! Peactivate Address Strobe
ASN = hi;
                                    ! Neactivale Lower Nata Strobe
LUSN = hi;
UUSN = hi;
                                    ! Deactivate Upper Data Strobe
                                    ! Are Placed Into Instruction
                                    ! Register
DITACKN = 61;
                                    ! Deactivate Data Transfer(Added)
                                    ! Acknowledge
next;
T = 5:
                                    ! Clock Cycle 5.
                                    ! Execute Frevious Assignment
next;
                                    ! Phase 1 Of
PHI1 = hi;
FHI2 = 10;
                                    ! Clack Cycle 5
RW = hi;
                                    ! Memory Read
                                    ! Disable Address Bus Buffer
ADENABLE = 10;
DBUS = Oxffff;
                                    ! Data Bus Returned To High
                                    ! Impedance State
                                   ! Disable Data Bus Buffer
DBENABLE = 10;
                                    ! Place Alil On Internal Address
labus = ADID:
                                    Hue
next;
                                    ! Execute Pending Assignments
                                    ! Phase 2 Of
PHI1 = lo;
                                    ! Clock Cycle 5
PH12 = hi;
ADENABLE = hi;
                                    ! Enable Address Bus Buffer
FCMODE = SRMODE:
                                    ! User Mode
                                    ! Accessing Program
FCSFACE = 1:
EXABUF = IABUS;
                                    ! Gate Internal Address Bus
                                    ! Place Low Word from ICID On
IDRUS = DILWORD;
                                    ! Internal Data Bus
                                    ! Into External Address Buffer
next:
ABUS = EXABUF;
                                    ! Address Placed On Bus(Added)
                                    ! Execute Pending Assignments
next;
T = 6;
                                    ! Clock Cycle 6
                                    ! Execute Assignment
next;
                                    ! Phase 1 Of
FHII = bi;
                                    ! Clock Cycle 6
PHI2 = 10;
                                    ! Assert Address Strobe
ASN = 10;
KW = 10;
                                    ! Place Contents Of Internal
EXDBUF = IDBUS;
                                    ! Data Bus Into External Data Buffer
                                    ! Reset Condition Code Bits
SRCARRY = 10;
SROVER = 10;
SRZEP0 = 1o:
SRNEG = 10:
                                    ! Execute Pending Assignments
next;
```

こうことには、国際の人の人の人の人の人を見ていていているとう。

```
! Phase 2
PHI1 = 10;
PHI2 = hi;
                                ! Of Clock Cycle 6
if EXDBUF eq1 0
                                ! Set Zero Condition Bit If Needed
  SRZERO = hi;
TOBUS = EXTIBUF;
                               ! Place Data On External Data Bus
                                / Enable Nata Bus
DBENABLE = hi;
next;
                                ! Execute Pending Assignments
<del>\*********************</del>
                                ! Clock Cycle 7
T = 7:
                                ! Execute Assignment
next;
                                ! Phase 1
PHI1 = hi;
                                ! Of Clock Cycle 7
PHI2 = 16;
if EXDBUF(15)
                               ! Set Negative Condition Bit
                               ! If Needed
  SRNEG = hi;
UTISN = lo;
                               ! Activate Upper And
LDSN = 10;
                                ! Lower Data Strobes
                                ! Wait Cycle Counter Initialized
twant = 0;
next;
                               ! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
     twait = twait + 1;
                               ! Increment Wait Cycle
     next;
                                ! Execute Impending Assignments
     PHI1 = lo;
                                ! Phuse 2
     PHI2 = hi:
                                ! Of Clock Cycle ?
                                ! Execute Assignments
     next;
     T = 8;
                                ! Clock Cycle 8
                                ! Execute Assignment
     next;
     PHI1 = hi;
                                ! Phase 1
     PHI2 = 10;
                                ! Of Clock Cycle 8
     if twait eql 2
                               ! Memory Responds After 2 Cycles
     MEABUSD = DBUS(15:8);
                               ! Store Data From Rus
     MEABUS + 13 = DBUS(7:0);
                               ! In Memory
     ICTACKN = 10
                               ! Asserts DTACKN(Added)
     next;
                               ! Execute Pending Assignments
     T = 7
                               ! Return To Phase 2
                               ! Of Clock Cycle 7
     );
                               ! Execute Impending Assignments
     next;
7 = 0;
                                ! Clock Cycle 8
                                ! Execute Assignment
next;
```

```
! Phase 2
     FHII = lo;
                                           ! Of Clock Cycle 8
     PHI2 = hi;
                                           ! Execute Pending Assignments
     next;
      人来来承担来来来,虽然这是世界来来,这些我就是我们的一个世界的人,也是这些人,也是是这些人,也是是这个人,也是这个人,也是这个人,也是这个人,我们就是这个人,我们就是这个人,
     T = 9;
                                            ! Clock Cycle 9
     next;
                                            ! Execute Assignment
     THII = hi;
                                            ! Phase 1
    . PHI2 = 10;
                                            ! Of Clock Cycle 9
     next;
                                           ! Execute Pending Assignments
     FHI1 = 10;
                                           ! Phase 2
                                            ! Of Clock Cycle 9
     PHI2 = hi;
                                            ! Deactivate Address Strobe
     ASN = ha;
                                            ! Deactivate Lower Data Strobe
     LDSN = hi;
     UDSN = hi;
                                           ! Deactivate Upper Data Strobe
                                           ! Increment Program Counter
     PC = PC + 2;
     IR = PFR:
                                           ! Flace Contents Of Frefetch
                                            ! Register Into Instruction
                                            ! Register
     DTACKN = hi;
                                            ! Deactivate Data Transfer
                                            ! Acknowledge(Added)
                                           ! Execute Pending Assignments
     next;
     T = 0
                                            ! JMF (A0)
ச்ஸ்≎ :=
      PHI1 = hi;
                                           ! Phase 1 Of
                                            ! Clock Cycle 0
     PH12 = 10;
                                           ! Place Data Bus In A High Impedance
     DBUS = 0xffff;
     RW = hi;
                                            ! Memory Read
     ADENABLE = 10;
                                            ! Disable Address Bus Buffer
     DIBENABLE = To;
                                            ! Disable Data Bus Buffer
     IMBUS = PC;
                                           ! Place PC On Internal Address
                                            ! Execute Pending Assignments
     next;
                                            ! Phase 2 Of
     FHI1 = 10;
                                            ! Clock Cycle O
     PHI2 = hi;
                                            ! Enable Address Bus Buffer
     ADENABLE = hi;
                                            ! Gate Internal Address Rus
     EXABUF = TABUS:
                                            ! Into External Address Buffer
                                            ! User Mode
     FCMODE = SRMODE;
     FOSPACE = 2;
                                            ! Accessing Program
     next;
                                            ! Execute fending Assignments
```

```
! Address Placed On Bus(Added)
ABUS = EXABUF;
                               ! Execute Pending Assignments
next;
T = 1:
                               ! Clock Cycle 1
                               ! Execute Assignment
next;
PHI1 = hi:
                               ! Phase 1 Of
FHI2 = lo;
                               ! Clock Cycle 1
ASN = lo:
                              ! Assert Address Strobe
LISN = 15;
                              ! Assert Lower Data Strobe
                              ! Assert Upper Data Strobe
UISN = 15:
                              ! Move Jump Address From AEOJ
CABUS = AEOD;
                              ! To Internal Address Buffer
                               ! Enable Data Bus
DEENABLE = ha:
                               ! Execute Fending Assignments
next;
                              ! Phase 2
FHI1 = 10;
                               ! Of Clock Cycle 1
PHI2 = hi;
                              ! Place Jump Address Into Program
FC = IABUS;
                               ! Counter
ne it;
T = 2:
                              ! Clock Cycle 2
next;
                               ! Execute Assignment
PHI1 = hi:
                               ! Of Clock Cycle 2
PHI2 = 16;
                              ! Wait For Memory To Place
while DTACKN eqt hi
                              ! Nata On The Bus
    next;
                              ! Execute Impending Assignments
     PHI1 = lo;
                              ! Phase 2
     PHI2 = hi;
                              ! Of Clock Cycle 2
     next;
                               ! Execute Assignments
     1 = 3:
                              ! Clock Cycle 3
                               ! Execute Assignment
     next:
     FHI1 = hi;
                              ! Phase 1
                               ! Of Clock Cycle 3
     PHI2 = 10;
     DRUSK15:89 = MEARUSI;
                              ! Memory Flaces Instruction
     DBUS(7:0) = MEADUS + 1];
                              ! On Data Rus And
                              ! Asserts DTACKN(Added)
     DTACKN = 10;
     next:
                               1 Execute Pending Assignments
     T = 2
                               ! Return To Phase 2:
                               ! Of Clock Cycle 2
    ):
     next;
                              ! Execute Impending Assignments
```

```
T = 3;
                                  ! Clock Cycle 3
next;
                                  ! Execute Assignment
FKII = 10;
                                  ! Phase 2
                                  ! Of Clock Cycle 3
PHI2 = hi;
EXHBUF = IN:US;
                                  ! Instruction On Data Bus
                                  ! Is Placed In External Data
                                  ! Bus Buffer
                                  ! Execute Fending Assignments
next;
T = 4;
                                  ! Clock Cycle 4
next:
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1
PHI2 = 10;
                                  ! Of Clock Cycle 4
next;
PFR = EXDBUF;
                                  ! The Contents Of The External
                                  ! Data Bus Buffer Are Placed
                                  ! In Prefetch Register
next;
                                  ! Execute Pending Assignments
f'Hii = lo;
                                  ! Phase 2
                                  ! Of Clock Cycle 4
PHI2 = hi;
                                  ! Neactivate Address Strobe
ASN = hi;
LUSN = hi;
                                  ! Deactivate Lower Nata Strobe
                                  ! Neactivate Upper Nata Strobe
UDSN = hi;
DTACKN = hi;
                                  ! Deactivate Data Transfer
                                  ! Acknowledge(Added)
T = 5:
                                  ! Clock Cycle 5
next;
                                  ! Execute Previous Assignment
PHI1 = hi;
                                  ! Phase 1 Of
FHI2 = 10;
                                  ! Clock Cycle 5
RW = hi;
                                  ! Memory Read
ADENABLE = 10;
                                  ! Disable Address Rus Buffer
                                  ! Disable Data Bus Buffer
DRENABLE = 10;
TABUS = PC;
                                  ! Place PC On Internal Address
next;
                                  ! Execute Pending Assignments
FiiI1 = Io;
                                  ! Phase 2 Of
PHI2 = hi;
                                  ! Clock Cycle 5
ADENABLE = h1;
                                  ! Enable Address Bus Buffer
                                  ! User Mode
FCMODE = SRMODE;
FCSPACE = 2;
                                  ! Accessing Program
                                  ! Gate Internal Address Bus
EXABUF = IABUS;
                                  ! Into External Address Buffer
next;
ABUS = EXABUF;
                                  ! Address Placed On Bus(Added)
```

```
! Execute Pending Assignments
next;
! Clock Cycle 6
next;
                              ! Execute Assignment
PHII = hi;
                             ! Phase 1 Of
FB12 = 10;
                             ! Clock Cycle 6
ASN = 10;
                              ! Assert Address Strobe
                             ! Assert Lower Data Strobe
LDSN = lo;
UDSN = 10;
                             ! Assert Upper Data Strobe
DBENABLE = hi:
                             ! Enable Data Rus
next;
                              ! Execute Pending Assignments
PHI1 = 10;
                              ! Phase 2
                             ! Of Clock Cycle 6
PH12 = hi;
next;
                              ! Execute Pending Assignments
T = 7;
                              ! Clock Cycle 7
next;
                              ! Execute Assignment
FHI1 = hi:
                             ! Phase 1
                             ! Of Clock Cycle 7
PHI2 = 10:
while DTACKN eql hi
                             ! Wait For Memory To Flace
                             ! Data On The Bus
    next;
                             ! Execute Impending Assignments
    PHI1 = 10;
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 7
    next;
                             ! Execute Assignments
    T = 8;
                             ! Clock Cycle 8
    next;
                             ! Execute Assignment
    PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 8
    PHI2 = 10;
    DBUS<15:8> = MEABUS3:
                             ! Memory Flaces Instruction
    DBUS<7:0> = MEABUS + 13;
                             ! On Data Bus And
    DTACKN = 10;
                             ! Asserts DTACKN(Added)
    nexti
                             ! Execute Pending Assignments
    T = 7
                             ! Return To Phase 2
                             1 Of Clock Cycle 7
    );
    next:
                             ! Execute Impending Assignments
T = 8:
                             ! Clock Cycle 8
next;
                              ! Execute Assignment
```

```
! Phase 2
     PHI1 = lo;
                                          ! Of Clock Cycle 8
     PHI2 = hi;
     EXDBUF = DBUS;
                                          ! Instruction On Data Bus
                                          ! Is Placed In External Data
                                          ! Bus Buffer
                                          ! Execute Pending Assignments
     next;
     T = 9;
                                          ! Clock Cycle 9
     next;
                                          ! Execute Assignment
     PHI1 = hi;
                                          ! Phase 1
                                          ! Of Clock Cycle 9
     PHI2 = 10;
                                          ! The Contents Of The External
     PFR = EXDBUF;
                                          ! Data Rus Buffer Are Placed
                                          ! In Frefetch Register
     next;
                                          ! Execute Pending Assignments
                                          ! Phase 2
     PHI1 = 10;
     PHI2 = hi;
                                          ! Of Clock Cycle 9
                                          ! Deactivate Address Strobe
     ASN = hi;
                                          ! Neactivate Lower Nata Strobe
     LIISN = hi;
     UDSN = hi;
                                          ! Deactivate Upper Data Strobe
     F'C = PC + 2;
                                          ! Increment Program Counter
                                          ! Place Contents Of Prefetch
     IR = PFR;
                                          ! Register Into Instruction
                                          ! Register
     DTACKN = hi:
                                          ! Deactivate Data Transfer
                                          ! Acknowledge(Added)
     next:
                                          ! Execute Pending Assignments
     T = 0
                                          ! Reset Clock Cycle Counter
decode execute_prefetch :=
                       case IR
                            031201: move
                                          ! MOVE.W D1,(A1)
                            047320: jmp
                                          ! JMP (AO) If IR = Octal Value
                       esac
                       )
main :=
    power_on_initialize;
    fetch_initial_instruction;
    while READY eql hi
          decode_execute_prefetch
     )
```

```
*/
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.L 11,A1 INSTRUCTION
                                                 */
/*
                                                 1/
/*
                                                 */
/*
                                                 1/
              Structure Declarations
                                                 */
/*
state
*/
/*
                                                 */
           M68000 Programming Registers
/*
                                                 */
DE0:73<31:0>.
                   ! 8 Data Registers
AE0:63<31:0>,
                   ! 7 Address Registers
UA7<31:0>.
                   ! User Stack Pointer
SA7<31:0>.
                   ! System Stack Pointer
PC<31:0>,
                   ! Program Counter
SR<15:0>,
                   ! Status Register
/*
                                                 */
/*
                                                 */
           Temporary Internal Registers
                                                 */
/*
PFR<15:0>,
                   ! Frefetch Register
IR<15:0>,
                   ! Instruction Register
FC<2:0>,
                   ! Function Code Register
EXDBUF<15:0>,
                   ! External Data Bus Buffer Register
EXABUF<23:1>,
                   ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,
                   ! ALU Buffer 1
ALUBUF2<31:0>,
                   ! ALU Buffer 2
DTEMP<15:0>.
                   ! Temporary Data Storage
                   ! Temporary Displacement Storage
DISREG<31:0>.
SRTEMP<15:0>,
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPAUR<31:0>,
                   ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>,
                   ! Temporary Access Type Storage
                   ! (Exception frocessing)
VECADR<23:0>,
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANAUR<31:0>.
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
T<7:0>,
                        ! Clock Cycle Counter
RESET.
                      ! Reset Flip-Flop
HALT,
                     ! Halt Flip-Flop
RW.
                     ! Read/Write Flip-Flop
ADENABLE,
                     ! Address Rus Buffer Enable
DBENABLE,
                      ! Data Bus Buffer Enable
ASN,
                     ! Address Strobe Flip-Flop
                      ! Lower Data Strobe Flip-Flop
LDSN,
                     ! Upper Data Strobe Flip-Flop
UDSN,
DITACKN.
                     ! Data Transfer Acknowledge Flip-Flop
COUT,
                     ! Carry Flip-Flop
EXCEPT,
                     ! Exception Processing Flip-Flop
READY.
                     ! Ready Flip-Flop
/x
                                                              */
                                                              */
/*
       Model transformation modifications:
/*
                                                              */
/*
           1) CDL decoder structure nonexistent in ISP' and un-
                                                              */
       necessary for model. Eliminated.
/*
                                                              */
/*
           Multi-phase clock structure nonexistent in ISP'.
                                                              */
/*
       Operations on registers will provide its equivalent.
                                                              */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
/*
       register will provide its equivalent.
                                                              */
           4) The declared bus structures are modeled with registers */
/*
/*
      without loss of model accurracy. This done to maintain model
                                                              */
/¥
       equivalency and simplicity.
                                                              */
/*
           5) The memory word length was reduced from 16 to 8 bit
                                                              */
/*
       words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*
       their PC incrementation, and to enable the use of existing
                                                              */
/*
      MC68000 assembler and linker/loader models. The memory was
                                                              */
/*
       also reduced from 8 Mwords to 32 Kbytes.
                                                              1/
                                                              */
/x
IABUS<31:0>,
                        ! Internal Address Rus
IDBUS<31:0>,
                        ! Internal Data Bus
                      ! Power Switch
SWITCH.
                      ! Phase 1 Of Two-Phase Clock
PHI1.
                      ! Phase 2 Of Two-Phase Clock
PHI2;
port
1/
/*
/*
             External Address and Data Bus
                                                              */
                                                              1/
/*
DBUS(15:0).
                        ! External Data Bus
ABUS<23:1>;
                        ! External Address Bus(changed)
```

```
/*
/*
                 Register Subfields
                                                           */
/*
PCADDR
          ≈ PC<23:0>,
                       ! Program Counter Address Field
SRTRACE
          ≈ SR<15>,
                       ! Trace Bit
SKMODE
          = SR<13>.
                        Mode Selection Bit
                         Carry Bit
SRCARRY
          ≃ SR<0>.
          = SR<1>,
SROVER
                       ! Overflow Rit
SRZERO
          ≈ SR<2>.
                       ! Zero Bit
          = SR<3>,
SRNEG
                        Negative Bit
SREX
          = SR<4>,
                        Extend Bit
SRMASK
          = SR<10:8>,
                         Interrupt Mask
                       ! Memory Access Address Space
          = FC<1:0>,
FCSFACE
FCHODE
          ≈ FC<2>,
                       ! User/Supervisor Mode Bit
                       ! FC Low Word
F'CLDW
          = PC<15:0>.
PCHI
          = PC<31:16>,
                       ! PC High Word
IJOLWORD
          = 1(0)<15:0>
                       ! DEOJ Low Word
DILWORD
          = BE1J<15:0>.
                       ! D[1] Low Word
          = DC23<15:0>,
                       ! DC23 Low Word
D2LWORD
DISLWORD
          = D[3](15:0),
                       ! DE31 Low Word
DIALWORD.
          = D[4]<15:0>,
                       ! II[4] Low Word
DSLWORD
                       ! U[5] Low Word
          = DE53<15:0>,
DIGLWORD
          = BE63<15:0>,
                       ! ILSJ Low Word
D7LWORD
                       ! DE73 Low Word
          = DE73<15:0>,
DISKEGHWORD = DISREG(31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
          = HANADR<15:0>, ! HANADR Low Word
HANADRLOW
HANADRHI
          = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI
          TEMPADR<31:16>;! TEMPADR High Word
Memory
/x
                                                           */
/*
                 16K 16-Bit Word Internal Memory
                                                           */
/x
ME0:327673<7:0>:
macro
/*
/*
                Logic Level Macros
                                                           */
/*
                                                           */
```

formut

```
10
     = 0 1,
hi
    = 1 %,
    = 0 1.
off
On
    = 1 %.
clear = 0 1;
                 *********************************
                                                          */
/* Fower On and Initialization. This process was not modeled but is
                                                          */
/*
   added to initialize signals and registers.
                                                          */
                                                          ¥/
power_on_initialize :=
                                  ! Turn Power On
      SWITCH = on;
                                  ! Execute Assignment
      next;
      READY = lo;
                                  ! System Not Ready
      RESET = lo;
                                  ! Assert Reset For
                                  ! 100 Miliseconds(Active Low)
      delay(100);
      RESET = hi;
                                  ! Deactivate Reset
                                  ! Execute Pending Assignments
      next;
      ASN = hi;
                                  ! Initialize Address Strobe
                                  ! Initialize Lower Nata Strobe
      LDSN = hi;
                                  ! Initialize Upper Data Strobe
      UIISN = hi;
      DTACKN = hi;
                                  ! Initialize Data Transfer Acknowledge
                                  ! Initialize Read/Write(Read On High)
      RW = hi;
      DBUS = Oxffff;
                                 ! Place Data Bus In High Impedance State
      M[0x100a] = 0xff;
                                  ! Place Memory Locations Following The
      M[0\times100b] = 0\times ff;
                                   ! JMP Instruction In A High State
      HALT = hi:
                                 ! Initialize Halt Flip-Flop(Active
                                  ! Low)
                                  ! Initialize Clock Cycle Counter
      T = 0:
      REALY = hi;
                                  ! System Ready
      /*
                                                          */
      /*
                                                          */
            Routine Initialization Per Hamby and Guillory
      /*
                                                          */
      ! Place Hex 5555555 Into D[1]
      D(1) = 0 \times 55555555;
                                  ! Place Hex 1004 Into ACO]
      A[0] = 0x1004;
      PC = 0x1000;
                                  ! Place Hex 1000 Into Program Counter
      next
                                  ! Execute Assignments
/*
                                                          1/
   Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                          */
/*
/* to retrieve modeled instructions for simulation and analysis. It
                                                          1/
   was fashsioned from the Read Cycle described by Hamby and Guillory */
```

on page VI-15 of their thesis.

```
/*
fetch_initial_instruction :=
     人本本於本本於本本於本於本來本本本本本本本本本或本本文於大於大於大於大於大於大學大學之文於大學本於本來本學大學之來不及其所不及於於
                                       ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                      ! Clock Cycle 0
                                      ! Memory Read
     RW = hi;
                                       ! Disable Address Bus Buffer
     ADENABLE = 10;
                                      ! Disable Data Bus Buffer
     DRENABLE = 10;
                                      ! Flace PC On Internal Address
     IABUS = PC;
                                      ! Bus
     next;
                                      ! Execute Pending Assignments
     PHI1 = 10;
                                      ! Phase 2 Of
     PHI2 = hi:
                                      ! Clock Cycle 0
     ADENABLE = hi;
                                      ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                      ! Gate Internal Address Bus
                                      ! Into External Address Buffer
     FCMODE = SRMODE;
                                      ! User Mode
     FCSFACE = 2;
                                      ! Accessing Program
                                      ! Execute Impending Assignments
     next;
     ABUS = EXABUF;
                                      ! Address Flaced On Bus(Added)
                                      ! Execute Pending Assignments
     next;
     T = 1:
                                      ! Clock Cycle 1
     next;
                                      ! Execute Assignment
                                      ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10:
                                      ! Clock Cycle 1
     ASN = lo;
                                      ! Assert Address Strobe
     L.IISN = 10;
                                      .! Assert Lower Data Strobe
     UDSN = 10:
                                      ! Assert Upper Data Strobe
     DBENABLE = hi;
                                      ! Enable Data Rus
                                      ! Execute Pending Assignments
     next:
                                      ! Phase 2
     PHI1 = 10;
     PHI2 = hi;
                                      ! Of Clock Cycle 1
     next;
                                      ! Execute Pending Assignments
     T = 2;
                                      ! Clock Cycle 2
     next;
                                      ! Execute Assignment
     PHI1 = hi;
                                      ! Phase 1
     PHI2 = 10;
                                      ! Of Clock Cycle 2
     while DTACKN eql hi
                                      ! Wait For Memory To Flace
                                      ! Data On The Bus
          next;
                                      ! Execute Impending Assignments
```

```
FHI1 = lo;
                                ! Phase 2
                                ! Of Clock Cycle 2
    FHI2 = hi;
                                ! Execute Assignments
    next;
     T = 3;
                                ! Clock Cycle 3
    next;
                                ! Execute Assignment
    PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 3
    PHI2 = 10;
     DBUS<15:8> = MEABUS];
                                ! Memory Flaces Instruction
    L(RUS<7:0) = MEARUS + 1];
                                ! On Data Bus And
     DTACKN = 10;
                                ! Asserts DTACKN(Added)
                                ! Execute Pending Assignments
    next:
     7 = 2
                                ! Return To Phase 2
                                ! Of Clock Cycle 2
    .);
    next;
                                ! Execute Impending Assignments
! Clock Cycle 3
T = 3;
next;
                                ! Execute Assignment
PHI1 = lo;
                                ! Phase 2
                                ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DRUS;
                                ! Instruction On Data Rus
                                ! Is Placed In External Data
                                ! Bus Buffer
                                ! Execute Pending Assignments
next;
! Clock Cycle 4
T = 4;
                                ! Execute Assignment
next;
PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 4
PHI2 = 10;
                                ! The Contents Of The External
PFR = EXDBUF;
                                ! Data Bus Buffer Are Flaced
                                ! In Frefetch Register
next;
                                ! Execute Fending Assignments
PHI1 = lo;
                                ! Phase 2
                                ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                                ! Deactivate Address Strobe
LOSN = hi;
                                ! Deactivate Lower Data Strobe
                                ! Deactivate Upper Data Strobe
UDSN = hi;
                                ! Contents Of Prefetch Register
IR = PFR:
                                ! Are Placed Into Instruction
                                ! Register
                                ! Deactivate Data Transfer(Added)
DTACKN = hi;
```

```
! Acknowledge
     PC = PC + 4;
                                           ! Increment Program Counter
     next:
                                           ! Execute Pending Assignments
     T = 0
                                           ! Reset Clock Cycle Counter
                                           ! AND.W #$DFFF,SR
andi :=
                                           ! Effect Of Instruction
     SAMODE = 10;
     IR<15:8> = MEPC3;
                                           ! Frefetch Next Instruction
     IR<7:0> = MEFC + 13;
                                           ! Is To Set Status Register
     next:
      PC = PC + 2;
                                              ! Increment Program Counter
    T = 5;
                                           ! Supervisor Bit To User
    next;
    T = 0
                                           ! Requires & Clock Cycles
                                           ! MOVE.L D1,A1
move :=
     PHI1 = hi;
                                           ! Phase 1 Of
                                           ! Clock Cycle 0
     PHI2 = 10;
                                           ! Disable Address Bus
     ADENABLE = 10;
     INTENABLE = 10:
                                           ! Disable Data Bus
     DBUS = 0xffff;
                                           ! Place Data Bus In High Impedance
                                           ! Memory Read
     RW = hi;
     IABUS = PC;
                                           ! Place PC On Internal Address
                                           ! Rus
     IDBUS = D[1]:
                                        ! Place Data From DC13 Onto
                                           ! Internal Data Bus
     next;
                                           ! Execute Pending Assignments
                                           ! Phase 2 Of
     PHI1 = lo;
                                           ! Clock Cycle 0
     PHI2 = hi;
                                           ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                           ! Gate Internal Address Bus
                                           ! Into External Address Buffer
     FCMODE = SRMODE;
                                           ! User Mode
     FCSPACE = 2;
                                           ! Accessing Program
                                           ! Clear Status Register Carry Bit
     SRCARRY = 10;
     SROVER = lo;
                                           ! Clear Status Register Overflow Bit
     SRZERO = lo;
                                           ! Clear Status Register Zero Bit
     SRNEG = 10;
                                           ! Clear Status Register Negative Bit
     A[1] = IDBUS;
                                        ! Place Data From Internal Data Bus
                                           ! Into A[1]
                                           ! Execute Impending Assignments
     next;
     ABUS = EXABUF;
                                           ! Address Flaced On Bus(Added)
     next;
                                           ! Execute Pending Assignments
```

```
! Clock Cycle 1
T = 1;
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1 Of
PHI2 = 10;
                                ! Clock Cycle 1
                                ! Assert Address Strobe
ASN = lo:
                                ! Assert Lower Data Strobe
LDSN = lo;
UDSN = lo;
                                ! Assert Upper Data Strobe
                                ! Enable Data Rus
DBENABLE = hi;
if A[1] eq1 0
                             ! Set Status Register Zero Bit
  SRZERO = hi;
                                ! If Moved Data Is Zero
                                ! Execute Pending Assignments
next;
PHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 1
if AE13<31>
                                ! Set Status Register Negative
  SRNEG = hi;
                                ! Bit If Moved Data Is Negative
                                ! Execute Pending Assignments
next;
T = 2;
                                ! Clock Cycle 2
                                ! Execute Assignment
next;
                                ! Phase 1
FHI1 = hi;
                                ! Of Clock Cycle 2
PHI2 = 10;
while DTACKN eql hi
                                ! Wait For Memory To Place
                                ! Data On The Bus
     next;
                                ! Execute Impending Assignments
     PHI1 = lo;
                                ! Phase 2
     PHI2 = hi;
                                ! Of Clock Cycle 2
                                ! Execute Assignments
     next;
     T = 3:
                               ! Clock Cycle 3
     next:
                                ! Execute Assignment
                                ! Phase 1
     PHI1 = hi;
     PHI2 = 10;
                                ! Of Clock Cycle 3
     DBUS<15:8> = MEABUS3;
                                ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                                ! On Data Bus And
     DTACKN = 10;
                                ! Asserts DTACKN(Added)
                                ! Execute Pending Assignments
     next;
     ! Return To Phase 2
     T = 2
                                ! Of Clock Cycle 2
     );
     next;
                               ! Execute Impending Assignments
T = 3:
                                ! Clock Cycle 3
                                ! Execute Assignment
next;
```

```
! Phase 2
     PHI1 = 10;
     PHI2 = hi;
                                           ! Of Clock Cycle 3
     EXDRUF = DBBS:
                                           ! Instruction On Data Bus
                                           ! Is Placed In External Data
                                           ! Bus Buffer
     next;
                                           ! Execute Pending Assignments
     人士并将式发出于其实这次来来中于文字来并更次来代史本北大大士的工术大士的一个二人之人的工作的工作的工作的工作的工作的工作的工作。
     T = 4;
                                           ! Clock Cycle 4
                                           ! Execute Assignment
     next;
     PHI1 = hi;
                                           ! Phase 1
     PHI2 = 10;
                                           ! Of Clock Cycle 4
     PFR = EXDRUF;
                                           ! The Contents Of The External
                                           ! Data Bus Buffer Are Flaced
                                           ! In Frefetch Register
     next;
                                           ! Execute Pending Assignments
     PHI1 = lo;
                                           ! Phase 2
     PHI2 = hi:
                                           ! Of Clock Cycle 4
     ASN = hi;
                                           ! Deactivate Address Strobe:
     LDSN = hi:
                                           ! Neactivate Lower Nata Strobe
     UDSN = hi;
                                           ! Deactivate Upper Data Strobe
     IR = PFR:
                                           ! Contents Of Prefetch Register
                                           ! Are Placed Into Instruction
                                           ! Register
     DTACKN = hi;
                                           ! Deactivate Data Transfer(Added)
                                           ! Acknowledge
     PC = PC + 2;
                                           ! Increment Program Counter
     next:
                                           ! Execute Impending Assignments
     T = 0
                                           ! Reset Clock Cycle Counter
                                           ! JMP (A0)
=: qmi
     PHI1 = hi;
                                           ! Phase 1 Of
     PHI2 = 10;
                                           ! Clock Cycle 0
     DBUS = Oxffff;
                                           ! Place Data Bus In A High Impedance
                                           ! Memory Read
     f:W = hi;
     ADENABLE = 10:
                                           ! Disable Address Bus Buffer
     DBENABLE = 10;
                                           ! Disable Data Bus Buffer
     IABUS = PC;
                                           ! Place FC On Internal Address
                                           ! Rus
                                           ! Execute Pending Assignments
     next;
     PHI1 = 10;
                                           ! Phase 2 Of
     PHI2 = bi:
                                           ! Clock Cycle O
     ADENABLE = hi:
                                           ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                           ! Gate Internal Address Bus
```

```
! Into External Address Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSFACE = 2;
                                 ! Accessing Program
                                 ! Execute Pending Assignments
next;
                                 ! Address Placed On Rus(Added)
ABUS = EXABUF:
                                 ! Execute Pending Assignments
next:
T = 1;
                                 ! Clock Cycle 1
                                 ! Execute Assignment
next;
FHI1 = hi;
                                 ! Phase 1 Of
                                 ! Clock Cycle 1
PHI2 = 10;
ASN = lo;
                                 ! Assert Address Strobe
                                 ! Assert Lower Data Strobe
LISN = lo:
                                 ! Assert Upper Data Strobe
UDSN = 10:
                                 ! Move Jump Address From A[3]
IABUS = A[0];
                                 ! To Internal Address Buffer
DBENABLE = hi;
                                 ! Enable Data Bus
next;
                                 ! Execute Pending Assignments
                                 ! Phase 2
FHI1 = 10;
FHI2 = hi;
                                 ! Of Clock Cycle 1
PC = IABUS:
                                 ! Place Jump Address Into Program
                                 ! Counter
next:
T = 2;
                                 ! Clock Cycle 2
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
PHI2 = 10;
                                 ! Of Clock Cycle 2
                                 ! Wait For Memory To Flace
while DTACKN eql hi
                                 ! Data On The Rus
                                 ! Execute Impending Assignments
     next;
     FHIII = lo;
                                 ! Phase 2
     PHI2 = hi;
                                 ! Of Clock Cycle 2
     next;
                                 ! Execute Assignments
     T = 3:
                                 ! Clock Cycle 3
     next;
                                 ! Execute Assignment
                                 ! Phase 1
     PHI1 = hi;
     PHI2 = 10;
                                 ! Of Clock Cycle 3
     DBUS<15:8> = MEABUS];
                                 ! Memory Flaces Instruction
     DBUS<7:0> = MCABUS + 13;
                                 ! On Data Bus And
                                 ! Asserts ITACKN(Added)
     INTACKN = 10;
                                 ! Execute Pending Assignments
     next;
```

(

```
T = 2
                                 ! Return To Phase 2
                                 ! Of Clock Cycle 2
     );
     next;
                                 ! Execute Impending Assignments
T = 3;
                                 ! Clock Cycle 3
next:
                                 ! Execute Assignment
PHI1 = 10:
                                 ! Phase 2
                                 ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DBUS;
                                 ! Instruction On Nata Bus
                                 ! Is Placed In External Data
                                 ! Bus Buffer
                                 ! Execute Pending Assignments
next;
T = 4;
                                 ! Clock Cycle 4
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
PHI2 = 10;
                                 ! Of Clock Cycle 4
next;
PFR = EXDBUF;
                                 ! The Contents Of The External
                                 ! Data Bus Buffer Are Placed
                                 ! In Prefetch Register
                                 ! Execute Pending Assignments
next;
FHI1 = lo;
                                 ! Phase 2
                                 ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi:
                                 ! Deactivate Address Strobe
LUSN = hi;
                                 ! Deactivate Lower Data Strobe
                                 ! Neactivate Upper Nata Strobe
UDSN = hi;
DTACKN = hi;
                                 ! Deactivate Data Transfer
                                 ! Acknowledge(Added)
next;
T = 5;
                                 ! Clock Cycle 5
                                 ! Execute Previous Assignment
next;
PHI1 = hi;
                                 ! Phase 1 Of
PH12 = 10;
                                 ! Clock Cycle 5
                                 ! Memory Read
RW = hi;
                                 ! Disable Address Bus Buffer
ADENABLE = lo;
DBENABLE = 10;
                                 ! Disable Data Bus Ruffer
                                 ! Place PC On Internal Address
IARUS = PC;
                                 ! Bus
next;
                                 ! Execute Pending Assignments
FHII = 10;
                                 ! Phase 2 Of
PH12 = hi:
                                 ! Clock Cycle 5
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                 ! User Mode
```

```
FCSPACE = 2;
                               ! Accessing Program
EXABUF = IABUS;
                                ! Gate Internal Address Bus
next;
                               ! Into External Address Ruffer
ABUS = EXABUF;
                               ! Address Flaced On Bus(Added)
next;
                               ! Execute Pending Assignments
! Clock Cycle 6
T = 6;
next;
                                ! Execute Assignment
PHI1 = hi;
                               ! Phase 1 Of
PHI2 = 10:
                               ! Clock Cycle 6
ASN = 10;
                               ! Assert Address Strobe
LDSN = lo;
                               ! Assert Lower Data Strobe
UDSN = lo;
                               ! Assert Upper Data Strobe
                               ! Enable Data Bus
DBENABLE = hi;
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 6
PHI2 = hi;
                               ! Execute Pending Assignments
next:
! Clock Cycle 7
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 7
PHI2 = lo;
                               ! Wait For Memory To Place
while ITACKN eql hi
                               ! Data On The Bus
                               ! Execute Impending Assignments
     next;
     PHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 7
     PHI2 = hi;
                               ! Execute Assignments
     next;
     T = 8;
                               ! Clock Cycle 8
     next;
                               ! Execute Assignment
     PHI1 = hi;
                               ! Plase 1
     PHI2 = lot
                               ! Of Clock Cycle 8
     DBUS<15:8> = MEABUS3;
                               ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                               ! On Nata kus And
     DTACKN = 10:
                               ! Asserts DTACKN(Added)
                               ! Execute Fending Assignments
     next;
     T = 7
                              ! Return To Phase 2
                              ! Of Clock Cycle 7
     );
     next;
                              ! Execute Impending Assignments
```

```
T = 8;
                                        ! Clock Cycle 8
     next;
                                        ! Execute Assignment
     PHI1 = lo;
                                        ! Phase 2
                                        ! Of Clock Cycle 8
     PHI2 = hi;
     EXDBUF = DBUS;
                                        ! Instruction On Data Bus
                                        ! Is Placed In External Data
                                        ! Bus Buffer
     next;
                                        ! Execute Pending Assignments
     T = 9:
                                        ! Clock Cycle 9
     next;
                                        ! Execute Assignment
                                        ! Phase 1
     PHI1 = hi;
     PHI2 = 10;
                                        ! Of Clock Cycle 9
                                        ! The Contents Of The External
     PFR = EXDRUF;
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
     next;
                                        ! Execute Pending Assignments
     PHI1 = lo;
                                        ! Phase 2
     PHI2 = hi:
                                        ! Of Clock Cycle 9
                                        ! Deactivate Address Strobe
     ASN = hi;
     LDSN = hi;
                                        ! Deactivate Lower Data Strobe
                                        ! Deactivate Upper Data Strobe
     UDSN = h1;
     PC = PC + 2;
                                        ! Increment Program Counter
                                        ! Place Contents Of Prefetch
     IR = PFR;
                                        ! Register Into Instruction
                                        ! Register
     DTACKN = hi:
                                        ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
     next;
                                        ! Execute Pending Assignments
     T = 0
                                        ! Reset Clock Cycle Counter
     )
decode_execute_prefetch :=
                     case IR .
                          0x2241: move
                                        ! MOVE.L DI,A1
                                        ! AND.W #$DFFF.SR
                          0x027c: andi
                                        ! JMP (A0) If IR = Octal Value
                          047320: jmp
                      esuc
                      )
main :=
    power_on_initialize;
    fetch_initial_instruction;
    while READY eql hi
         decode_execute_prefetch
```

大人一章では他的なのは、 これのはなる。

```
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W D1, (A1)+ INSTRUCTION
                                               */
/*
/×
/x
/x
              Structure Declarations
                                               */
/*
                                               */
state
/*
/*
           M68000 Programming Registers
                                               */
/x
DE0:73<31:0>,
                  ! 8 Data Registers
A[0:6]<31:0>,
                  ! 7 Address Registers
UA7<31:0>,
                  ! User Stack Pointer
SA7<31:0>,
                  ! System Stack Pointer
PC<31:0>,
                  ! Program Counter
SR<15:0>,
                  ! Status Register
/*
                                               */
/*
           Temporary Internal Registers
                                               */
/*
                                               */
PFR<15:0>,
                   ! Prefetch Register
IR<15:0>,
                  ! Instruction Register
FC<2:0>.
                  ! Function Code Register
                  ! External Data Bus Buffer Register
EXDBUF<15:0>,
EXABUF<23:1>,
                  ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,
                  ! ALU Buffer 1
ALUBUF2<31:0>.
                   ! ALU Buffer 2
DITEMP<15:0>,
                   ! Temporary Data Storage
DISREG<31:0>,
                    Temporary Displacement Storage
                    Temporary Status Register Storage
SRTEMP<15:0>,
                   ! (Exception Processing)
                    Temporary Instruction Register Storage
IRTEMP<15:0>.
                   ! (Exception Processing)
TEMPADR<31:0>.
                   ! Temporary Cycle Address Storage
                    (Exception Processing)
ACTYPE<15:0>,
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
                   ! Temporary Vector Address Storage
VECADR<23:0>,
                   ! (Exception Processing)
```

```
HANADR<31:0>.
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
T<7:0>,
                        ! Clock Cycle Counter
RESET.
                      ! Reset Flip-Flop
HALT,
                      ! Halt Flip-Flop
FiW,
                      ! Read/Write Flip-Flop
ADENABLE,
                       Address Bus Buffer Enable
DIBENABLE,
                      ! Data Bus Buffer Enable
ASN,
                      ! Address Strobe Flip-Flop
LIISN.
                      ! Lower Data Strobe Flip-Flop
                      ! Upper Data Strobe Flip-Flop
UDISN,
DITACKN.
                      ! Data Transfer Acknowledge Flip-Flop
                      ! Carry Flip-Flop
COUT.
EXCEPT,
                      ! Exception Processing Flip-Flop
READY.
                      ! Ready Flip-Flop
/x
                                                              1/
/*
       Model transformation modifications:
                                                              */
1*
                                                              */
/*
                                                              */
           1) CDL decoder structure nonexistent in ISP' and un-
       necessary for model. Eliminated.
11
                                                              */
/*
           Multi-phase clock structure nonexistent in ISF'.
                                                              */
       Operations on registers will provide its equivalent.
                                                              */
/*
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
/*
       register will provide its equivalent.
                                                              */
/#

 The declared bus structures are modeled with registers ★/

       without loss of model accurracy. This done to maintain model
                                                              */
/*
                                                              */
11
       equivalency and simplicity.
/*
           5) The memory word length was reduced from 16 to 8 bit
                                                              */
/*
       words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*
       their PC incrementation, and to enable the use of existing
                                                              1/
       MC68000 assembler and linker/loader models. The memory was
                                                              */
/*
       also reduced from 8 Mwords to 32 Kbytes.
                                                              1/
/*
/*
                                                              */
       ! Internal Address Bus
1ABUS<31:0>,
IDBUS<31:0>,
                        ! Internal Data Bus
                        ! Wait State Counter
twait<4:0>.
SWITCH,
                      ! Power Switch
FHI1,
                      ! Phase 1 Of Two-Phase Clock
                      ! Phase 2 Of Two-Phase Clock
PHI2;
port
/*
                                                              */
                                                              */
/*
             External Address and Data Bus
/*
DBUS(15:0),
                        ! External Data Bus
```

(O

```
ABUS<23:1>;
                     ! External Address Bus(changed)
format
/*
                                                         */
                                                         */
/*
                Register Subfields
/*
                                                         */
PCADDR
         = PE<23:0>,
                      ! Program Counter Address Field
SRTRACE
         = SR<15>,
                      ! Trace Bit
SKMODE
         = 'SR<13>,
                      ! Mode Selection Bit
SRCARRY
         = SR<0>,
                      ! Carry Bit
SROVER
                      ! Overflow Bit
         = S8<1>,
SRZERO
         = SR<2>.
                      ! Zero Bit
SRNEG
         = SR<3>,
                      ! Negative Bit
SREX
                      ! Extend Bit
         = SR<4>,
SRMASK
         = SR<10:8>,
                      ! Interrupt Mask
FICSPIACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCMODE
                      ! User/Supervisor Mode Bit
         = FC<2>.
F'CLOW
         = PC<15:0>,
                      ! PC Low Word
PCHI
         = PC(31:16),
                      ! PC High Word
DOLWORD
         = DE03<15:0>.
                      ! II[O] Low Word
DILWORD
         = DE13<15:0>,
                      ! [IE1] Low Word
D2LWORD
         = DC23<15:0>,
                      ! DE23 Low Word
         = D[3]<15:0>,
D3LWORD
                      ! DE33 Low Word
DALWORD
         = DE43<15:0>,
                      ! DE41 Low Word
DSLWORD
         = DE53<15:0>,
                      ! DES3 Low Word
II6LWORD
         = DE63<15:0>,
                      ! D[6] Low Word
D7LWORD
                      ! BE73 Low Word
         = DE73<15:0>,
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
         = HANADR<15:0>. ! HANADR Low Word
HANADIRLOW
HANADEHI
         = HANADR<31:16>,! HANADR High Word
TEMPAURLOW = TEMPAUR(15:0), ! TEMPAUR Low Word
         = TEMPADR<31:16>;! TEMPADR High Word
TEMPADRHI
memory
*/
/*
                                                         */
                 16K 16-Bit Word Internal Memory
                                                         */
/*
ME0:327473<7:0>;
BIGCTO
/*
                                                         */
/*
                Logic Level Macros
```

```
10
    = 0 %.
    = 1 2,
ħi
off
     = 0 %.
οn
    = 1 %,
clear = 0 %;
/*
/* Power On and Initialization. This process was not modeled but is
                                                         X/
                                                         */
  added to initialize signals and registers.
                                                         */
/ X
power_on_initialize :=
      SWITCH = on;
                                 ! Turn Power On
                                 ! Execute Assignment
      next;
                                 ! System Not Ready
      READY = lo:
                                 ! Assert Reset For
      RESET = lo;
                                 ! 100 Miliseconds(Active Low)
      delay(100);
      RESET = hi;
                                 ! Deactivate Reset
                                 ! Execute Pending Assignments
      next;
                                 ! Initialize Address Strobe
      ASN = hi;
                                 ! Initialize Lower Data Strobe
      LDSN = hi;
                                 ! Initialize Upper Data Strobe
      UDSN = hi;
                                 ! Initialize Nata Transfer Acknowledge
      DITACKN = hi;
      RW = hi:
                                 ! Initialize Read/Write(Read On High)
      DBUS = 0 \times ffff;
                                ! Flace Data Bus In High Impedance State
      MEO \times 100cJ = O \times ff;
                                 ! Place Memory Locations Following The
                                  ! JMP Instruction In A High State
      MEO \times 100 d3 = 0 \times ff;
                                 ! Initialize Halt Flip-Flop(Active
      HALT = hi;
                                 ! Low)
                                 ! Initialize Clock Cycle Counter
      T = 0;
      READY = hi;
                                 ! System Ready
      */
      /*
      /*
            Routine Initialization Fer Hamby and Guillory
                                                         X/
      /*
                                                         */
      ! Place Hex 5555 Into D[1]
      DE13 = 0.85555;
      D[2] = 0:2000;
                              ! Will Be Used To Reset A[1]
      A[0] = 0x1004;
                                 ! Place Hex 1004 Into A[0]
      A[1] = 0 \times 2000;
                              ! Store Nata At This Address
                                 ! Place Hex 1000 Into Frogram Counter
      PC = 0x1000;
                                 ! Execute Assignments
      next
      )
Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                         */
```

```
/* to retrieve modeled instructions for simulation and analysis. It
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
  on page VI-15 of their thesis.
                                                            */
/*
                                                            */
fetch_initial_instruction :=
     PHI1 = hi;
                                     ! Phase 1 Of
     PHI2 = 10;
                                     ! Clock Cycle 0
     RW = hi;
                                     ! Memory Read
     ADENABLE = 10;
                                     ! Disable Address Bus Buffer
     IIBENABLE = 10;
                                     ! Disable Data Bus Buffer
     IARUS = PC;
                                     ! Flace FC On Internal Address
     next;
                                     ! Execute Pending Assignments
     PHI1 = lo;
                                     ! Phase 2 Of
     PHI2 = hi;
                                     ! Clock Cycle 0
     ADENABLE = hi;
                                     ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                     ! Gate Internal Address Bus
                                     ! Into External Address Buffer
     FCMODE = SRMODE;
                                     ! User Mode
     FCSPACE = 2;
                                     ! Accessing Program
                                     ! Execute Impending Assignments
     next;
     ABUS = EXABUF;
                                     ! Address Placed On Bus(Added)
     next;
                                     ! Execute Pending Assignments
     T = 1:
                                     ! Clock Cycle 1
     next;
                                     ! Execute Assignment
     PHI1 = hi;
                                     ! Phase 1 Of
     FHI2 = 10;
                                     ! Clock Cycle 1
                                     ! Assert Address Strobe
     ASN = 10;
     LDSN = 10;
                                     ! Assert Lower Data Strobe
     UDSN = lo:
                                     ! Assert Upper Data Strobe
     DENABLE = hi;
                                     ! Enable Data Bus
     next;
                                     ! Execute Pending Assignments
     FHI1 = lo;
                                     ! Phase 2
     PHI2 = hi:
                                     ! Of Clock Cycle 1
     next;
                                     ! Execute Fending Assignments
     T = 2;
                                     ! Clock Cycle 2
     next;
                                     ! Execute Assignment
                                     ! Phase 1
     PHI1 = hi;
     PHI2 = 10;
                                     ! Of Clock Cycle 2
```

```
! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
    next:
                               ! Execute Impending Assignments
    FHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 2
     PHI2 = hi;
     next;
                               ! Execute Assignments
     T = 3:
                               ! Clock Cycle 3
                               ! Execute Assignment
    next;
                               ! Phase 1
    PHI1 = h1;
    PHI2 = 10;
                               ! Of Clock Cycle 3
                               ! Memory Places Instruction
     DBUS(15:8> = MEABUS];
    DRUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
     DTACKN = 10;
                               ! Asserts DTACKN(Added)
    next;
                               ! Execute Fending Assignments
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
    );
    next;
                               ! Execute Impending Assignments
T = 3;
                               ! Clock Cycle 3
next;
                                ! Execute Assignment
FHI1 = lo:
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Sycle 3
EXDRUF = DEUS:
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
next;
                               ! Execute Pending Assignments
J = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PHIL = hi;
                               ! Phase 1
PH12 = 10;
                               ! Of Clock Cycle 4
                               ! The Contents Of The External
PFR = EXDBUF;
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 4
ASN = hi;
                               ! Deactivate Address Strobe
LDSN = hi:
                               ! Beactivate Lower Data Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
IR = PFR;
                               ! Contents Of Prefetch Register
```

```
! Are Placed Into Instruction
                                        ! Register
     DTACKN = hi;
                                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
     PC = PC + 4;
                                        ! Increment Program Counter
                                        ! Execute Pending Assignments
     next:
                                        ! Reset Clock Cycle Counter
     I = 0
andi :=
                                        ! AND.W #$DFFF,SR
    SRMODE = 10;
                                        ! Effect Of Instruction
    IR<15:8> = MEPC3;
                                        ! Prefetch Next Instruction
    1R<7:0> = MEPC + 13;
                                        ! Is To Set Status Register
      PC = PC + 2;
                                           ! Increment Program Counter
    T = 5;
                                        ! Supervisor Rit To User
                                        ! Mode
    next:
    T = 0
                                        ! Requires 6 Clock Cycles
                                          ! MOVE.W D1,(A1)+
moveinc :=
     PHI1 = hi;
                                        ! Phase 1 Of
                                        ! Cluck Cycle 0
     PHI2 = 10;
     DBUS = 0xffff;
                                        ! Place Data Bus In High Impedance
                                        ! Memory Read
     F:W = hi:
     ADENABLE = 10;
                                        ! Disable Address Bus Buffer
     ABUS = 0xffffff;
                                        ! Address Bus High Impedanced
                                        ! Disable Data Bus Buffer
     DBENABLE = 10;
     TABUS<31:1> = PC<31:1>;
                                        ! Place PC On Internal Address
                                        ! Bus
     next;
                                        ! Execute Pending Assignments
                                        ! Phase 2 Of
     PHI1 = lo;
     PHI2 = hi;
                                        ! Clock Cycle 0
     ADENABLE = hi;
                                        ! Enable Address Bus Buffer
     EXABUF = IABUS<23:1>;
                                        ! Gate Internal Address Bus
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSPACE = 2;
                                        ! Accessing Program
     ABUS = IABUS(23:1>;
                                        ! Address Placed On Bus
                                        ! Execute Impending Assignments
     next;
     T = 1;
                                        ! Clock Cycle 1
     next;
                                        ! Execute Assignment
     PHI1 = bi;
                                        ! Phase 1 Of
     PHI2 = 10;
                                        ! Clock Cycle 1
```

```
! Assert Address Strobe
ASN = lo;
LDSN = lo;
                               ! Assert Lower Data Strobe
UIISN = lo;
                               ! Assert Upper Data Strobe
DBENABLE = hi;
                               ! Enable Data Bus
next;
                               ! Execute Pending Assignments
PHI1 = lo:
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 1
                               ! Execute Pending Assignments
next;
T = 2;
                               ! Clock Cycle 2
                               ! Execute Assignment
next;
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 2
                               ! Wait For Memory To Flace
while DTACKN eql hi
                               ! Bata On The Rus
                               ! Execute Impending Assignments
     next;
     FHI1 = lo;
                               ! Phase 2
     FHI2 = hi;
                               ! Of Clock Cycle 2
                               ! Execute Assignments
     next;
     T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
     next;
     PHI1 = hi;
                               ! Phase 1
     PHI2 = lo:
                               ! Of Clock Cycle 3
     DBUS<15:E> = MCABUS];
                               ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
     DITACKN = 10;
                               ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
     next;
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
                               ! Execute Impending Assignments
     next;
T = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
                               ! Phase 2
PHI1 = lo;
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DRUS;
                               ! Instruction On Nata Rus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
```

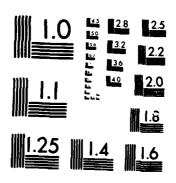
```
T = 4;
                                   ! Clock Cycle 4
next;
                                   ! Execute Assignment
                                   ! Phase 1
FHI1 = hi;
                                   ! Of Clock Cycle 4
FHI2 = lo;
                                   ! The Contents Of The External
PFR = EXDBUF;
                                   ! Data Bus Buffer Are Placed
                                   ! In Frefetch Register
                                   ! Execute Pending Assignments
next;
PHI1 = 10;
                                   ! Phase 2
PHI2 = hi:
                                   ! Of Clock Cycle 4
ASN = hi;
                                   ! Deactivate Address Strobe
LDSN = hi;
                                   ! Deactivate Lower Data Strobe
UDSN = hi:
                                   ! Deactivate Upper Data Strobe
                                   ! Are Flaced Into Instruction
                                   ! Register
DTACKN = hi;
                                   ! Neactivate Nata Transfer(Added)
                                   ! Acknowledge
next;
T = 5;
                                   ! Clock Cycle 5
next;
                                   ! Execute Previous Assignment
PHI1 = hi;
                                   ! Phase 1 Of
                                   ! Clock Cycle 5
PHI2 = 10;
RW = hi;
                                   ! Memory Read
ADENABLE = 10;
                                   ! Disable Address Bus Buffer
ABUS = Uxffffff;
                                   ! Address Bus High Impedanced
DBUS = 0xffff;
                                   ! Data Bus Returned To High
                                   ! Impedance State
DRENABLE = 10;
                                   ! Disable Data Bus Buffer
IABUS = AC13;
                                   ! Place A[1] On Internal Address
                                   ! Rus
next;
                                   ! Execute Pending Assignments
PHI1 = lo;
                                   ! Phase 2 Of
                                   ! Clock Cycle 5
PHI2 = hi;
                                   ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                   ! User Mode
FCSPaCE = 1;
                                   ! Accessing Program
EXABUF = IABUS<23:1>;
                                   ! Gate Internal Address Bus
                                   ! Place Low Word from D[1] On
IDBUS = DILWORD;
                                   ! Internal Data Bus
ABUS = IABUS;
                                   ! Flace Address On Bus
next;
                                   ! Into External Address Buffer
T = 6;
                                   ! Clock Cycle 6
next;
                                   ! Execute Assignment
FHI1 = hi;
                                   ! Phase 1 Of
```

```
! Clock Cycle 6
PHI2 = 10;
                                  ! Assert Address Strobe
ASN = 10;
RW = 10;
EXDBUF = IDBUS;
                                  ! Place Contents Of Internal
                                  ! Nata Rus Into External Nata Buffer
                                  ! Reset Condition Code Bits
SACARRY = 10;
SROVER = 10;
SRZERO = 10;
SRNEG = lo:
next;
                                 ! Execute Fending Assignments
PHI1 = lo;
                                  ! Phase 2
PHI2 = hi;
                                  ! Of Clock Cycle 6
if EXDBUF eql 0
                                 ! Set Zero Condition Bit If Needed
  SRZERO = hi;
                                 ! Place Data On External Data Bus
DRUS = EXDBUF;
                                 ! Enable Data Bus
DBENABLE = hi;
                                 ! Execute Pending Assignments
next;
T = 7:
                                  ! Clock Cycle 7
                                  ! Execute Assignment
next;
PHI1 = hi;
                                 ! Phase 1
                                 ! Of Clock Cycle 7
PHI2 = 10;
if EXDBUF(15)
                                 ! Set Negative Condition Bit
                                 ! If Needed
  SRNEG = hi;
UIISN = 10;
                                 ! Activate Upper And
                                 ! Lower Data Strobes
LDSN = lo;
                                 ! Wait Cycle Counter Initialized
twait = 0;
next;
                                 ! Wait For Memory To Place
while ITACKN eql hi
                                 ! Data On The Bus
                                 ! Increment Wait Cycle
     twait = twait + 1;
                                 ! Execute Impending Assignments
     next;
     PHI1 = 10;
                                  ! Phase 2
     PHI2 = h1;
                                 ! Of Clock Cycle 7
                                 ! Execute Assignments
     T = 8;
                                  ! Clock Cycle 8
     next;
                                 ! Execute Assignment
                                 ! Phase 1
     FHI1 = hi;
                                 ! Of Clock Cycle B
     PHI2 = 10:
                                 ! Memory Responds After 2 Cycles
     if twait eal 2
                                 ! Store Data From Bus
     MEARUS] = DRUS<15:8>;
                                 ! In Memory
     MEABUS + 1] = DBUS<7:0>;
     DITACKN = 10
                                 ! Asserts DTACKN(Added)
     );
     next;
                                 ! Execute Pending Assignments
```

```
T = 7
                                   ! Return To Phase 2
                                   ! Of Clock Cycle 7
         );
                                   ! Execute Impending Assignments
         next;
    T = 8:
                                    ! Clock Cycle 8
    next;
                                    ! Execute Assignment
    FHI1 = lo;
                                    ! Phase 2
    PHI2 = hi;
                                    ! Of Clock Cycle 8
    next;
                                    ! Execute Pending Assignments
     ! Clock Cycle 9
    T = 9;
    next;
                                    ! Execute Assignment
    PHI1 = bi:
                                    ! Phase 1
    PHI2 = 16:
                                    ! Of Clock Cycle 9
    A[1] = A[1] + 2;
                                    ! Increment AC13
                                    ! Execute Pending Assignments
    next;
    FHI1 = 10;
                                    ! Phase 2
                                    ! Of Clock Cycle 9
    PHI2 = hi;
    ASN = hi;
                                    ! Deactivate Address Strobe
    LDSN = hi;
                                    ! Deactivate Lower Data Strobe
    UDSN = hi;
                                    ! Neactivate Upper Data Strobe
    PC = PC + 2;
                                    ! Increment Program Counter
                                    ! Place Contents Of Prefetch
    IR = PFR;
                                    ! Register Into Instruction
                                    ! Register
    DTACKN = hi;
                                    ! Deactivate Data Transfer
                                    ! Acknowledge(Added)
                                    ! Execute Pending Assignments
    next;
    T = 0
                                    ! MOVE.L D2,A1
BOVE :=
     PHI1 = hi;
                                    ! Phase 1 Of
    PHI2 = 10;
                                    ! Clock Cycle 0
    ADENABLE = 10;
                                    ! Disable Address Bus
    IBENABLE = lo;
                                    ! Disable Data Bus
    INUS = Oxffff;
                                    ! Place Data Bus In High Impedance
    RW = hi;
                                    ! Memory Read
                                    ! Place PC On Internal Address
    IAHUS = PC;
                                    ! Rus
    IDBUS = DC23;
                                 ! Place Data From DC2J Onto
```

```
! Internal Data Bus
                                   ! Execute Pending Assignments
next:
                                   ! Phase 2 Of
FHI1 = lo;
                                   ! Clock Cycle O
PHI2 = hi;
ABENABLE = hi;
                                   ! Enable Address Bus Buffer
                                   ! Gate Internal Address Bus
EXABUF = IABUS;
                                   ! Into Extérnal Address Buffer
FCMODE = SRMODE;
                                   ! User Mode
FOSPACE = 2:
                                   ! Accessing Program
                                   ! Clear Status Register Carry Bit
SRCARRY = lo;
                                   ! Clear Status Register Overflow Bit
SROVER = 10;
                                   ! Clear Status Register Zero Bit
SRZERO = lo:
SRNEG = lo;
                                   ! Clear Status Register Negative Bit
                                 ! Flace Data From Internal Data Bus
AC13 = IBBUS;
                                   ! Into A[1]
                                   ! Execute Impending Assignments
next;
                                   ! Address Placed On Bus(Added)
ARUS = EXABUF;
                                   ! Execute Pending Assignments
next;
T = 1;
                                   ! Clock Cycle 1
next;
                                   ! Execute Assignment
PHI1 = hi;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 1
                                   ! Assert Address Strobe
ASN = lo:
                                   ! Assert Lower Data Strobe
LISN = lo;
UDSN = lo:
                                   ! Assert Upper Data Strobe
                                   ! Enable Data Bus
DBENABLE = hi;
                               ! Set Status Register Zero Bit
if A[1] eql 0
                                   ! If Moved Data Is Zero
  SRZERO = hi;
next:
                                   ! Execute Pending Assignments
PHI1 = lo;
                                   ! Phase 2
PHI2 = hi:
                                   ! Of Clock Cycle 1
if AE13<31>
                                   ! Set Status Register Negative
  SRNEG = hi;
                                   ! Bit If Moved Data Is Negative
                                   ! Execute Pending Assignments
! Clock Cycle 2
T = 2:
                                   ! Execute Assignment
next;
FHI1 = hi;
                                   ! Phuse 1
                                   ! Of Clock Cycle 2
PHI2 = 10:
                                   ! Wait For Memory To Place
while DTACKN eq1 hi
                                   ! Data On The Rus
                                   ! Execute Impending Assignments
     next;
                                   ! Phase 2
     PHI1 = lo;
                                   ! Of Clock Cycle 2
     PHI2 = hi;
     next;
                                   ! Execute Assignments
```

THE SIMULATION AND ANALYSIS OF A RTL HODEL OF THE MOTOROLA NGESSES NICROP. (U) AIR FORCE INST OF TECH MRIGHT-PATTERSON AFB ON SCHOOL OF ENGI. C A BAXLEY DEC 84 AFIT/GCS/ENG/84D-2-VOL-2 F/G 9/2 AD-A164 257 2/5 UNCLASSIFIED NL



THE SECOND DESCRIPTION OF THE SECOND DESCRIP

MICROCOPY RESOLUTION TEST CHART

```
T = 3;
                                ! Clock Cycle 3
    next;
                                ! Execute Assignment
     PHI1 = hi:
                                ! Phase 1
                                ! Of Clock Cycle 3
    FHI2 = 10;
    DBUS(15:8> = MEABUS];
                                ! Memory Places Instruction
                               ! On Data Bus And
    DBUS<7:0> = MEABUS + 13;
     DTACKN = lo:
                                ! Asserts DTACKN(Added)
    next;
                                ! Execute Pending Assignments
     T = 2
                                ! Return To Phase 2
                                ! Of Clock Cycle 2
     );
    next;
                               ! Execute Impending Assignments
T = 3;
                                ! Clock Cycle 3
                                ! Execute Assignment
next;
FHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 3
EXDBUF = DBUS;
                                ! Instruction On Data Rus
                                ! Is Placed In External Data
                                ! Rus Buffer
                                ! Execute Fending Assignments
next;
T = 4;
                                ! Clock Cycle 4
next;
                                ! Execute Assignment
                                ! Phase 1
PHI1 = hi;
PHI2 = 10:
                                ! Of Clock Cycle 4
                                ! The Contents Of The External
PFR = EXDRUF:
                                ! Data Bus Buffer Are Placed
                                ! In Prefetch Register
                                ! Execute Fending Assignments
nexti
PHI1 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 4
ASN = hi;
                                ! Deactivate Address Strobe
                                ! Neactivate Lower Nata Strobe
LUSN = hi;
                                ! Deactivate Upper Data Strobe
UDSN = hi;
IR = PFR;
                                ! Contents Of Prefetch Register
                                ! Are Placed Into Instruction
                                ! Register
                                ! Deactivate Data Transfer(Added)
DITACKN = hi;
                                ! Acknowledge
PC = PC + 2;
                                ! Increment Program Counter
                                ! Execute Impending Assignments
next;
                                ! Reset Clock Cycle Counter
T = 0
```

```
! JMF (A0)
=: نوس ز
     ! Phase 1 Of
     PHI1 = hi;
                                      ! Clock Cycle 0
     FHI2 = 10;
                                      ! Pluce Data Bus In A High Impedance
     DBUS = 0xffff;
                                      ! Memory Read
     RW = hi
     ADENABLE = 10;
                                      ! Disable Address Bus Buffer
     INTENABLE = 10;
                                      ! Disable Data Rus Ruffer
     IABUS = PC;
                                      ! Place PC On Internal Address
     next;
                                      ! Execute Pending Assignments
                                      ! Phase 2 Of
     PHI1 = lo;
     PHI2 = hi:
                                      ! Clock Cycle 0
                                      ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                      ! Gate Internal Address Bus
                                      ! Into External Address Buffer
     FCMODE = SRMODE;
                                      ! User Mode
     FCSFACE = 2;
                                      ! Accessing Program
     next;
                                      ! Execute Pending Assignments
     ARUS = EXARUF;
                                      ! Address Flaced On Bus(Added)
     next;
                                      ! Execute Pending Assignments
     ! Clock Cycle 1
     T = 1:
                                      ! Execute Assignment
     next;
     PHI1 = hi;
                                      ! Phase 1 Of
                                      ! Clock Cycle 1
     PHI2 = 10;
     ASN = 10;
                                      ! Assert Address Strobe
                                      ! Assert Lower Data Strobe
     LDSN = lo;
                                      ! Assert Upper Data Strobe
     UDSN = lo;
     IARUS = A[O];
                                      ! Move Jump Address From ACO]
                                      ! To Internal Address Buffer
     DBENABLE = hi:
                                      ! Enable Data Rus
     next:
                                      ! Execute Pending Assignments
     FHI1 = lo;
                                      ! Phase 2
                                      ! Of Clock Cycle 1
     PHI2 = hi;
     PC = IABUS;
                                      ! Place Jump Address Into Program
                                       ! Counter
     next;
     T = 2;
                                      ! Clock Cycle 2
     next;
                                      ! Execute Assignment
     PHI1 = hi;
                                      ! Phase 1
```

```
PHI2 = 10;
                              ! Of Clock Cycle 2
while DTACKN eql hi
                              ! Wait For Memory To Place
                               ! Nata On The Bus
    next;
                               ! Execute Impending Assignments
    f'HI1 = lc:
                               ! Phase 2
    PH12 = h1;
                               ! Of Clock Cycle 2
                               ! Execute Assignments
    next;
     T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
    next;
    PHI1 = h1;
                              ! Phase 1
                               ! Of Clock Cycle 3
    PHI2 = 10:
                               ! Memory Places Instruction
    DRUS (15:8) = MEARUS];
    DBUS(7:0) = MEABUS + 13;
                              ! On Data Bus And
    DITACKN = 10:
                               ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
    next;
    T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
    );
    next;
                              ! Execute Impending Assignments
\******<
                               !- Clock Cycle 3
7 = 3;
                               ! Execute Assignment
next;
FHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Rus Huffer
                               ! Execute Pending Assignments
next;
T = 4;
                               ! Clock Cycle 4
nexti
                               ! Execute Assignment
                               ! Phase 1
FHI1 = hi:
                               ! Of Clock Cycle 4
PHI2 = 10;
next;
FFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Frefetch Register
                               ! Execute Pending Assignments
next;
FHI1 = 10;
                               ! Phase 2
PHI2 = hi:
                               ! Of Clock Cycle 4
                               ! Deactivate Address Strobe
ASN = hi;
                               ! Deactivate Lower Data Strobe
LUSN = hi;
```

```
UIISN = hi;
                                 ! Deactivate Upper Data Strobe
                                 ! Neactivate Nata Transfer
ICTACKN = h1;
                                 ! Acknowledge(Added)
T = 5:
                               ! Clock Cycle 5
next;
                                 ! Execute Previous Assignment
PHI1 = hi:
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 5
RW = hi;
                                 ! Memory Read
ADENABLE = lo;
                                 ! Disable Address bus Buffer
                                 ! Disable Data Bus Buffer
DIRENABLE = 10;
1ABUS = PC;
                                 ! Place FC On Internal Address
                                 ! Execute Pending Assignments
next;
PHI1 = lo;
                                 ! Phase 2 Of
PH12 = hi;
                                 ! Cluck Cycle 5
                                 ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                 ! User Mode
FOSPACE = 2:
                                 ! Accessing Program
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
                                 ! Into External Address Buffer
nexti
                                 ! Address Flaced On Bus(Added)
ARUS = EXABUF;
next:
                                 ! Execute Pending Assignments
T = 6:
                                 ! Clock Cycle 6
                                 ! Execute Assignment
next;
FHI1 = hi;
                                 ! Phase 1 Of
FHI2 = 10:
                                 ! Clock Cycle &
ASN = 10:
                                 ! Assert Address Strobe
LIISN = 10;
                                 ! Assert Lower Nata Strobe
UDSN = lo;
                                 ! Assert Upper Data Strobe
IIBENABLE = hi;
                                 ! Enable Nata Rus
                                 ! Execute Pending Assignments
next:
FHI1 = lo;
                                 ! Phase 2
                                 ! Of Clock Cycle 6
PHI2 = hi;
                                 ! Execute Fending Assignments
next:
T = 7:
                                 ! Cluck Cycle 7
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
PH12 = 10;
                                 ! Of Clock Cycle 7
                                 ! Wait For Memory To Place
while DTACKN eql hi
                                 ! Data On The Rus
                                 ! Execute Impending Assignments
     next;
```

```
PHI1 = 10;
                                ! Phase 2
    PHI2 = ha:
                               ! Of Clock Cycle 7
                               ! Execute Assignments
     next;
     ! Clock Cycle 8
     next:
                                ! Execute Assignment
     FHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 8
    PHI2 = 10;
     DBUS<15:8/ = MEABUSJ;
                               ! Memory Flaces Instruction
    DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
     DYACKN = 10;
                                ! Asserts DTACKN(Added)
                               ! Execute fending Assignments
     next;
     ! Return To Phase 2
     T = 7
                               ! Of Clock Cycle 7
     );
     next;
                               ! Execute Impending Assignments
T = 8;
                                ! Clock Cycle 8
next;
                               ! Execute Assignment
f'HI1 = lo;
                                ! Phase 2
PHI2 = hi:
                               ! Of Clock Cycle 8
                               ! Instruction On Data Bus
EXIMUF = DBUS;
                               ! Is flaced In External Data
                               ! Bus kuffer
next;
                                ! Execute Fending Assignments
T = 9;
                               ! Clock Cycle 9
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                                ! Of Clock Cycle 9
FHI2 = 10;
PFR = EXDBUF;
                               ! The Contents Of The External
                                ! Data Bus Buffer Are Flaced
                               ! In Prefetch Register
next;
                                ! Execute Pending Assignments
PHI1 = lo:
                               ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 9
ASN = hi;
                                ! Deactivate Address Strobe
LIISN = hi;
                                ! Neactivate Lower Nata Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
FC = PC + 2;
                               ! Increment Program Counter
                               ! Place Contents Of Prefetch
IR = PFR:
                                ! Register Into Instruction
                                ! Register
                                ! Deactivate Data Transfer
DITACKN = hi;
```

```
! Acknowledge(Added)
                                               ! Execute Pending Assignments
      next;
                                               ! Reset Clock Cycle Counter
      T = 0
decode_execute_prefetch :=
                         case IR
                               0x2242: move
                                               ! MOVE.L D2,A1
                               0x32c1: moveinc! MOVE.W D1,(A1)+
                                              ! AND.W ##DFFF,SR
                               Ox027c: andi
                               047320: Jap
                                               ! JMP (AO) If IR = Octal Value
                         esac
main :=
     power_on_initialize;
     fetch_initial_instruction;
while READY eql hi
           decode_execute_prefetch
```

```
11
                                                x/
                                                */
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W D1,04(A1) INSTRUCTION
14
                                                */
/*
                                                */
/*
                                                */
              Structure Declarations
/*
                                                */
state
/*
/*
                                                */
           M68000 Programming Registers
/*
                                                */
DE0:73<31:0>.
                   ! 8 Data Registers
ALO:63<31:0>,
                   ! 7 Address Registers
U67<31:0>,
                  ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Pointer
PC<31:0>,
                   ! Program Counter
SR<15:0>.
                   ! Status Register
/*
                                                */
/*
           Temporary Internal Registers
                                                */
/*
PFR<15:0>,
                   ! Prefetch Register
IR<15:0>.
                   ! Instruction Register
FC<2:0>,
                   ! Function Code Register
EXDBUF<15:0>,
                   ! External Data Bus Buffer Register
                   ! External Address Bus Buffer Register(changed)
EXABUF<23:1>.
ALUBUF 1 < 31:0>,
                   ! ALU Ruffer 1
                   ! ALU Buffer 2
ALUBUF2<31:0>,
                   ! Temporary Data Storage
DITEMP<15:02.
DISREG<31:0>,
                   ! Temporary Displacement Storage
                   ! Temporary Status Register Storage
SRTEMP<15:0>,
                   ! (Exception Processing)
IRTEMP<15:0>,
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPADR<31:0>,
                   ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>,
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>.
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:0>.
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
                        ! Clock Cycle Counter
T<7:0>,
RESET.
                     ! Reset Flip-Flop
HALT.
                     ! Halt Flip-Flop
FiW,
                     ! Read/Write Flip-Flop
ADENABLE.
                     ! Address Bus Buffer Enable
                     ! Nata Bus Buffer Enable
DENABLE.
ASN.
                     ! Address Strobe Flip-Flop
                      ! Lower Data Strobe Flip-Flop
LIISN.
UDSN.
                     ! Upper Data Strobe Flip-Flop
INTACKN,
                     ! Data Transfer Acknowledge Flip-Flop
COUT,
                      ! Carry Flip-Flop
EXCEFT,
                      ! Exception Processing Flip-Flop
REALLY,
                      ! Ready Flip-Flop
/*
/*
      Model transformation modifications:
                                                              */
/*
                                                              */
/*
           1) CDL decoder structure nonexistent in ISP' and un-
                                                              */
/x
      necessary for model. Eliminated.
                                                              1/
/×
           Multi-phase clock structure nonexistent in ISP'.
                                                              */
/*
      Operations on registers will provide its equivalent.
                                                              */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
/*
       register will provide its equivalent.
                                                              */
/*
           4) The declared bus structures are modeled with registers */
       without loss of model accurracy. This done to maintain model
/*
                                                              */
/*
      equivalency and simplicity.
                                                              */
/*
           5) The memory word length was reduced from 16 to 8 bit
                                                              11/
      words to coincide with the ECR's 32-Kbyte memory, to agree with*/
/*
/*
       their PC incrementation, and to enable the use of existing
                                                              */
/*
      MC68000 assembler and linker/loader models. The memory was
                                                              */
       also reduced from 8 Mwords to 32 Kbytes.
/*
                                                              */
/*
                                                              */
IABUS<31:0>,
                        ! Internal Address Bus
IDBUS<31:0>,
                        ! Internal Data Bus
twait(4:0).
                        ! Wait State Counter
SWITCH.
                     ! Power Switch
PH11.
                     ! Phase 1 Of Two-Phase Clock
PHI2;
                     ! Phase 2 Of Two-Phase Clock
port
/*
                                                              */
/*
                                                              */
             External Address and Data Dus
/*
                                                              */
DBUS<15:0>.
                        ! External Data Bus
```

```
ABUS<23:1>:
                      ! External Address Bus(changed)
format
/*
                                                         */
/¥
                 Register Subfields
                                                         */
/1
                                                         x/
             PCADDR
         = PC(2310),
                      ! Frogram Counter Address Field
SRTRACE
         = SR<15>.
                      ! Trace Bit
         = SR<13>,
SKMODE
                      ! Mode Selection Bit
SRCARRY
                      ! Carry Bit
         = SR<0>,
                      ! Overflow Bit
SROVER
         = SR<1>,
SKZEKO
         = SR<2>,
                      ! Zero Bit
SKNEG
         = SR<3>,
                      ! Negative Bit
SREX
         = SR<4>,
                      ! Extend Bit
SRMASK
         = SR(10:8),
                      ! Interrupt Mask
FCSPACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCMUDE
         = FC<2>.
                      ! User/Supervisor Mode Bit
FICLOW
         = FC<15:0>,
                      ! PC Low Word
PCHI
         = PC<31:16>,
                      ! PC High Word
DOLWORD
         = 1/C03<15:0>,
                      ! DEOJ Low Word
DILWORD
         = I([1](15:0),
                      ! D[1] Low Word
I/2LWORD
         = DE23<15:0>,
                      ! DC23 Low Word
D3LW( 1
         = DE33<15:0>,
                      ! D[3] Low Word
II4LWÜR!
         = B[4](15:0),
                      ! DE43 Low Word
DELWORD
         = DC53<15:0>,
                      ! DES] Low Word
DISCHORD
         = DE63<15:0>,
                      ! DE63 Low Word
D7LWORD
         = DE73<15:0>,
                      ! DE73 Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANALIRLOW
        = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANAUR<31:16>,! HANAUR High Word
TEMPAURLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPAURHI
         = TEMPADR<31:16>;! TEMPADR High Word
Memory
*/
/*
/*
                 16K 16-Pit Word Internal Memory
                                                         */
                                                         */
/x
ME0:327673<7:0>;
BIGETS
/*
/*
                Logic Level Macros
                                                         */
```

```
= 0 1,
lū
ħi
    = 1 %,
    = 0 1,
off
on
     = 1 %,
clear = 0 %;
/×
/* Power On and Initialization. This process was not modeled but is
                                                          */
  added to initialize signals and registers.
                                                          */
18
14
                                                          */
power_on_initialize :=
      SWITCH = on;
                                 ! Turn fower On
      next:
                                 ! Execute Assignment
      READY = 10;
                                 ! System Not Ready
      RESET = lo;
                                 ! Assert Reset For
                                 ! 100 Miliseconds(Active Low)
      delay(100);
      RESET = hi:
                                 ! Deactivate Reset
                                 ! Execute Pending Assignments
      next;
      ASN = hi;
                                 ! Initialize Address Strobe
      LIBN = hi:
                                 ! Initialize Lower Nata Strobe
      UDSN = ha;
                                 ! Initialize Upper Nata Strobe
                                 ! Initialize Nata Transfer Acknowledge
      DITACKN = hi;
      BW = bi;
                                 ! Initialize Read/Write(Read On High)
      DBUS = Oxffff;
                                 ! Flace Nata Bus In High Impedance State
      M[0\times100e] = 0\times ff;
                                  ! Place Memory Locations Following The
      ME0x100f] = 0xff;
                                   ! JMF Instruction In A High State
      HALT = hi;
                                 ! Initialize Halt Flip-Flop(Active
                                 ! Low)
      T = 0;
                                 ! Initialize Clock Cycle Counter
      READY = hi;
                                 ! System Ready
      /*
                                                          */
      /*
            Routine Initialization Fer Hamby and Guillory
                                                          1/
      /*
                                                          */
      ! Place Hex 5555 Into DC13
      D[1] = 0x5555;
      A[0] = 0x1004;
                                  ! Place Hex 1004 Into A[0]
      A[1] = 0x2000;
                               ! Store Data At This Address
      FC = 0x1000:
                                 ! Place Hex 1000 Into Program Counter
      next
                                 ! Execute Assignments
*/
   Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                          */
/* to retrieve modeled instructions for simulation and analysis. It
```

```
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/# on page VI-15 of their thesis.
                                                           */
fetch_initial_instruction :=
     ! Phase 1 Of
    PHI1 = hi;
     PHI2 = 10:
                                     ! Clock Cycle 0
     RW = hi;
                                     ! Memory Read
     ADENABLE = 10;
                                     ! Disable Address Bus Buffer
     DRENABLE = 10;
                                     ! Disable Data Bus Buffer
     IABUS = PC;
                                     ! Flace FC On Internal Address
                                     ! Execute Fending Assignments
    next;
                                     ! Phase 2 Of
     PHI1 = lo;
                                    ! Clock Cycle 0
     PHI2 = hi;
                                     ! Enuble Address Bus Buffer
     MDENABLE = hi;
     EXABUF = TABUS;
                                     ! Gate Internal Address Rus
                                    ! Into External Address Buffer
     FCMODE = SRMODE:
                                     ! User Mode
     FCSPACE = 2;
                                     ! Accessing Program
     next:
                                    ! Execute Impending Assignments
     ABUS = EXABUF:
                                     ! Address Flaced On Bus(Added)
     next:
                                     ! Execute Fending Assignments
     T = 1;
                                     ! Clock Cycle 1
     next;
                                     ! Execute Assignment
                                     ! Phase 1 Of
     PHI1 = hi;
                                     ! Clock Cycle 1
     FHI2 = 10:
                                     1 Assert Address Strobe
     ASN = 10;
                                    ! Assert Lower Data Strobe
     LIISN = 10;
                                    ! Assert Upper Data Strobe
     UIISN = 10:
     DBENABLE = hi;
                                     ! Enable Data Rus
     next;
                                     ! Execute Pending Assignments
     FHI1 = lo;
                                     ! Phase 2
                                     ! Of Clock Cycle 1
     PHI2 = hi;
                                    ! Execute Fending Assignments
     next;
     ! Clock Cycle 2
     T = 2:
                                     ! Execute Assignment
     next;
     FHI1 = hi;
                                     ! Phase 1
                                     ! Of Clock Cycle 2
     PHI2 = 10;
     while DTACKN eql hi
                                     ! Wait For Memory To Place
```

```
! Data On The Bus
                               ! Execute Impending Assignments
    next;
    PHI1 = 10;
                               ! Phase 2
    PHI2 = hi;
                               ! Of Clock Cycle 2
                               ! Execute Assignments
    next:
     ! Clock Cycle 3
    T = 3;
                               ! Execute Assignment
    next;
                               ! Phase 1
    PHI1 = hi;
    FHI2 = 10;
                               ! Of Clock Cycle 3
                               ! Memory Places Instruction
    DBUS<15:8> = MEABUS];
                               ! On Data Bus And
    I(BUS <7:0) = MEARUS + 11;
    DTACKN = lo;
                               ! Asserts DTACKN(Added)
                               ! Execute Fending Assignments
    next;
     ! Return To Phase 2
    T = 2
                               ! Of Clock Cycle 2
    );
                               ! Execute Impending Assignments
    next;
! Clock Cycle 3
T = 3;
                               ! Execute Assignment
next;
                               ! Phase 2
PHI1 = lo:
PHI2 = hi;
                               ! Of Clock Cycle 3
EXDBUF = DBUS;
                               ! Instruction On Nata Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next:
! Clock Cycle 4
T = 4;
                               ! Execute Assignment
next;
FHI1 = bi:
                               ! Phase 1
                               ! Of Clock Cycle 4
FHI2 = 10;
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Nata Bus Buffer Are Flaced
                               ! In Prefetch Register
                               ! Execute Pending Assignments
next;
FHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 4
PHI2 = bi:
ASN = hi;
                               ! Deactivate Address Strobe
LUSN = hi;
                               ! Neactivate Lower Nata Strobe
                               ! Deactivate Upper Data Strobe
UDSN = hi;
IR = PFR;
                               ! Contents Of Prefetch Register
                                ! Are Placed Into Instruction
```

```
! Register
     DTACKN = hi;
                                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
     PC = PC + 4;
                                        ! Increment Program Counter
     next:
                                        ! Execute Fending Assignments
     T = 0
                                        ! Reset Clock Cycle Counter
     )
andi :=
                                        ! AND.W #$DFFF,SR
    SRMODE = 10:
                                        ! Effect Of Instruction
    IR<15:8> = MCPC3;
                                        ! Prefetch Next Instruction
    IR(7:0) = MEPC + 13;
                                        ! Is To Set Status Register
    next;
      PC = PC + 2;
                                           ! Increment Program Counter
    T = 5:
                                        ! Supervisor Bit To User
                                        ! Mode
    next;
    T = 0
                                        ! Requires 6 Clock Cycles
move :=
                                        ! MOVE.W D1.4(A1) [8(A1)]
     ! Phase 1 Of
     PHI1 = hi;
     FHI2 = 10;
                                        ! Clock Cycle 0
     DBUS = Oxffff;
                                        ! Place Note Bus In High Impedance
                                        ! Memory Read
     RW = hi;
     ADENABLE = 10;
                                        ! Disable Address Bus Buffer
     ABUS = Oxffffff;
                                        ! Address Rus High Impedanced
                                        ! Disable Data Bus Buffer
     DBENABLE = 10;
     IABUS<31:1> = PC<31:1>;
                                        ! Flace FC On Internal Address
                                        ! Rus
     next;
                                        ! Execute Pending Assignments
                                        ! Phase 2 Of
     PHI1 = lo;
     PHI2 = hi;
                                        ! Clock Cycle 0
                                        ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = labus<23:1>;
                                        ! Gate Internal Address Bus
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSPACE = 2;
                                        ! Accessing Program
     ABUS = IABUS<23:1>;
                                        ! Address Placed On Bus
     next:
                                        ! Execute Impending Assignments
     T = 1:
                                        ! Clock Cycle 1
     next;
                                        ! Execute Assignment
                                        ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                        ! Clock Cycle 1
                                        ! Assert Address Strobe
     ASN = 10;
```

```
LIISN = lo:
                             ! Assert Lower Data Strobe
UIISN = lo;
                            ! Assert Upper Data Strobe
DBENABLE = hi:
                            ! Enable Data Rus
next;
                            ! Execute Pending Assignments
PHI1 = lo:
                             ! Phase 2
FHI2 = hi;
                             ! Of Clock Cycle 1
                            ! Execute Pending Assignments
next;
T = 2;
                            ! Clock Cycle 2
                            ! Execute Assignment
next;
PHI1 = hi;
                            ! Phase 1
                            ! Of Clock Cycle 2
PHI2 = 10;
                            ! Wait For Memory To Place
while DTACKN eql hi
                            ! Data On The Bus
                            ! Execute Impending Assignments
    next;
                            ! Phase 2
    F'HI1 = lo;
    PHI2 = hi;
                            ! Of Clock Cycle 2
    next;
                            ! Execute Assignments
    T = 3;
                            ! Clock Cycle 3
                             ! Execute Assignment
    next;
    FHI1 = hi:
                            ! Phase 1
                            ! Of Clock Cycle 3
    PHI2 = 10;
    DBUS<15:8> = MEABUS3;
                            ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                            ! On Data Bus And
    INTACKN = 10;
                            ! Asserts DTACKN(Added)
    next:
                            ! Execute Pending Assignments
    T = 2
                             ! Return To Phase 2
                             ! Of Clock Cycle 2
    );
                            ! Execute Impending Assignments
    next;
T = 3;
                             ! Clock Cycle 3
next;
                             ! Execute Assignment
PHI1 = lo;
                             ! Phase 2
PHI2 = hi;
                            ! Of Clock Cycle 3
EXDBUF = DBUS;
                            ! Instruction On Data Rus
                             ! Is flaced In External Data
                             ! Bus Buffer
next;
                             ! Execute Pending Assignments
! Clock Cycle 4
```

```
! Execute Assignment
next;
                                  ! Phase 1
FHI1 = hi;
                                  ! Of Clock Cycle 4
FHI2 = 10;
DISREG = EXDBUF sxt 32;
                                  ! Store Displacement
next;
                                  ! Execute Pending Assignments
PHI1 = lo;
                                  ! Phase 2
                                  ! Of Clock Cycle 4
PHI2 = hi:
                                  ! Deactivate Address Strobe
ASN = hi;
LISN = hi:
                                  ! Reactivate Lower Rata Strobe
UDSN = hi;
                                  ! Deactivate Upper Data Strobe
                                  ! Are Flaced Into Instruction
                                  ! Register
PC = PC + 2;
                                  ! Increment Program Counter
DISREG = DISREG + ALIJ;
                             ! Add Address Register To Displacement
DTACKN = hi;
                                 ! Deactivate Data Transfer(Added)
                                  ! Acknowledge
next:
T = 5;
                                  ! Clock Cycle 5
                                  ! Execute Previous Assignment
next;
                                  ! Phase 1 Of
PHI1 = hi;
                                  ! Clock Cycle 5
PHI2 = 10;
                                  ! Memory Read
RW = hi;
                                 ! Disable Address Bus Buffer
ADENABLE = lo;
ABUS = 0xffffff;
                                 ! Address Bus High Impedanced
DBUS = 0xffff;
                                 ! Nata Rus Returned To High
                                  ! Impedance State
                                  ! Disable Data Rus Ruffer
DRENABLE = 10;
                               ! Place PC On Internal Address
IABUS(31:1> = PC(31:1>;
next;
                                  ! Execute Pending Assignments
                                  ! Phase 2 Of
PHI1 = 10;
                                  ! Clock Cycle 5
PHI2 = hi:
                                 ! Enable Address Bus Buffer
ADENABLE = hi;
                                 ! User Mode
FCMODE = SRMODE;
FCSPACE = 2:
                                 ! Accessing Data
                                 ! Gate Internal Address Bus
EXABUF = labus<23:10;
ARUS = IABUS<23:1>;
                                  ! Place Address On Bus
                                  ! Into External Address Buffer
next;
! Clock Cycle 6
T = 6;
                                  ! Execute Assignment
next;
PHI1 = hi;
                                  ! Phase 1 Of
                                  ! Clock Cycle 6
PHI2 = 10;
UDSN = lo:
                                  ! Activate Upper And
LIISN = lo;
                                  ! Lower Data Strobes
```

```
ASN = 10;
                             ! Assert Address Strobe
DBENABLE = hi;
                             ! Enable Data Bus
                            ! Execute Pending Assignments
next:
PHI1 = lo:
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 6
next;
                             ! Execute Pending Assignments
T = 7;
                             ! Clock Cycle 7
next;
                             ! Execute Assignment
PHI1 = hi;
                             ! Phase 1 Of
PHI2 = 10;
                             ! Clock Cycle 7
while DTACKN eql hi
                            ! Wait For Memory To Flace
                            ! Nata On The Bus
                            ! Execute Impending Assignments
    next;
    FHI1 = 10:
                            ! Phase 2
    PHI2 = hi;
                            ! Of Clock Cycle 7
                            ! Execute Assignments
    next;
    T = 8:
                            ! Clock Cycle 8
    next;
                            ! Execute Assignment
    PHI1 = hi:
                             ! Phase 1
                            ! Of Clock Cycle 8
    PH12 = 10;
    DBUS(15:8) = MCABUS];
                            ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                            ! On Data Bus And
                            ! Asserts IMACKN(Added)
    DITACKN = 10;
                            ! Execute Pending Assignments
    next;
    T = 7
                            ! Return To Phase 2
                            ! Of Clock Cycle 7
    );
    next;
                            ! Execute Impending Assignments
1 = 8;
                            ! Clock Cycle 8
next;
                             ! Execute Assignment
fHI1 = lo;
                            ! Phase 2
PHIC = hi;
                            ! Of Clock Cycle 8
                             ! Instruction On Data Rus
EXDBUF = DBUS:
                             ! Is flaced In External Data
                            ! Bus Buffer
next;
                             ! Execute Pending Assignments
T = 9;
                            ! Clock Cycle 9
                             ! Execute Assignment
next;
```

```
! Phase 1
PHI1 = hi;
                                    ! Of Clock Cycle 9
f'H12 = 10;
                                    ! The Contents Of The External
PFR = EXDBUF;
                                    ! Data Bus Buffer Are Placed
                                    ! In Frefetch Register
nexti
                                    ! Execute Pending Assignments
PHI1 = 10;
                                    ! Phase 2
PHI2 = hi;
                                    ! Of Clock Cycle 9
ASN = hi;
                                    ! Deactivate Address Strobe
LISN = hi:
                                    ! Neactivate Lower Nata Strobe
                                    ! Deactivate Upper Data Strobe
UDSN = hi:
                                    ! Deactivate Data Transfer(Added)
IITACKN = hi:
                                    ! Acknowledge
next;
                                    ! Execute Pending Assignments
T = 10;
                                     ! Clock Cycle 10
next;
                                    ! Execute Assignment
FHI1 = hi;
                                    ! Phase 1 Of
PHI2 = 10;
                                    ! Clock Cycle 10
DBUS = 0xffff;
                                    ! Place Nata Bus In High Impedance
                                    ! Memory Read
RW = hi;
                                    ! Disable Address Bus Buffer
ADENABLE = 10;
ABUS = 0xffffff;
                                    ! Address Bus High Impedanced
                                    ! Disable Data Bus Buffer
DRENABLE = 16;
                                    ! Place DISREG On Internal Address
IABUS = DISREG;
next;
                                    ! Execute Pending Assignments
                                    ! Phase 2 Of
PHI1 = 1c;
                                    ! Clock Cycle 10
PHI2 = hi:
                                    ! Enable Address Bus Buffer
ADENABLE = hi:
                                    ! Gate Internal Address Rus
EXABUF = IABUS<23:1>;
                                    ! Into External Address Buffer
                                    ! Flace Low Word Of DE13 On Bus
IDBUS = DILWORD;
FCMODE = SRMODE;
                                    ! User Mode
FCSPACE = 1;
                                    ! Accessing Data
ABUS = IABUS<23:1>;
                                    ! Address Flaced On Bus
                                    ! Execute Impending Assignments
next;
*! Clock Cycle 11
T = 11;
next:
                                    ! Execute Assignment
PHI1 = hi;
                                    ! Phase 1 Of
FHI2 = 10;
                                    ! Clock Cycle 11
ASN = 10;
                                    ! Assert Address Strobe
RW = lo;
EXDEUF = IDEUS;
                                    ! Flace Contents Of Internal
                                    ! Data Bus Into External Data Buffer
```

```
SRCARRY = 10;
                                 ! Reset Condition Code Bits
SROVER = 10;
SRZERO = lo;
SRNEG = 10;
                                ! Execute Pending Assignments
next;
PHI1 = 10;
                                 ! Phase 2
                                 ! Of Clock Cycle 11
FHI2 = hi;
                                 ! Set Zero Condition Bit If Needed
if EXDBUF eq1 0
  SRZERO = hi;
DRUS = EXDRUF;
                                ! Place Nata On External Nata Bus
DBENABLE = hi;
                                 ! Enable Data Bus
                                 ! Execute Pending Assignments
next;
T = 12;
                                 ! Clock Cycle 12
next:
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
PHI2 = 10;
                                 ! Of Clock Cycle 12
                                ! Set Negative Condition Bit
if EXDBUF<15>
  SRNEG = hi:
                                ! If Needed
UDSN = 10;
                                ! Activate Upper And
LIISN = 10;
                                ! Lower Data Strobes
twait = 0;
                                ! Wait Cycle Counter Initialized
next:
                                ! Wait For Memory To Place
while INTACKN eql hi
                                ! Data On The Bus
  . (
     twait = twait + 1;
                                ! Increment Wait Cycle
     next;
                                1. Execute Impending Assignments
     FHI1 = 10;
                                ! Phase 2
     PHI2 = hi:
                                ! Of Clock Cycle 12
     next;
                                ! Execute Assignments
     T = 13:
                                 ! Clock Cycle 13
                                ! Execute Assignment
     next;
     FHI1 = hi;
                                ! Phase 1
     PHI2 = 10;
                                ! Of Clock Cycle 13
     if twait eql 2
                                ! Memory Responds After 2 Cycles
     MEABUS] = DBUS<15:8>;
                                ! Store Data From Bus
     MCABUS + 1] = DBUS<7:0>;
                                ! In Memory
     DITACKN = 10
                                ! Asserts DTACKN(Added)
     );
     next;
                                ! Execute Pending Assignments
     T = 12
                                ! Return To Phase 2
                                ! Of Clock Cycle 12
     );
```

```
! Execute Impending Assignments
          next;
     T = 13;
                                       ! Clock Cycle 13
                                      ! Execute Assignment
     next;
     PHI1 = 10;
                                      ! Phase 2
     PHI2 = hi;
                                      ! Of Clock Cycle 13
                                      ! Execute Pending Assignments
     next;
     T = 14;
                                       ! Clock Cycle 14
     next;
                                      ! Execute Assignment
     PHI1 = hi;
                                      ! Phase 1
                                      ! Of Clock Cycle 14
     PHI2 = 10;
     next:
                                      ! Execute Pending Assignments
                                      ! Phase 2
     f'HI1 = lo;
                                      ! Of Clock Cycle 9
     PHI2 = hi;
     ASN = hi;
                                      ! Neactivate Address Strobe
     LDSN = hi;
                                      ! Neactivate Lower Nata Strobe
     UDSN = hi;
                                       ! Neactivate Upper Nata Strobe
     PC = PC + 2;
                                       ! Increment Program Counter
                                       ! Place Contents Of Frefetch
     IR = PFR;
                                      ! Register Into Instruction
                                      ! Register
     DITACKN = hi;
                                       ! Deactivate Data Transfer
                                       ! Acknowledge(Added)
     next;
                                       ! Execute Pending Assignments
     T = 0
                                       ! JMP (AO)
=: qmi.
     ! Phase 1 Of
     PHI1 = hi;
                                       ! Clock Cycle 0
     PHI2 = 10:
                                       ! Place Data Bus In A High Impedance
     DBUS = 0xffff;
     RW = hi;
                                       ! Memory Read
                                      ! Disable Address Bus Buffer
     ADENABLE = lo;
                                       ! Disable Data Bus Buffer
     DIBENABLE = 10;
     IABUS = PC;
                                      ! Place PC On Internal Address
                                       ! Rus
     next;
                                      ! Execute Pending Assignments
                                      ! Phase 2 Of
     PHI1 = lo;
                                      ! Clock Cycle 0
     PHI2 = hi;
     ADENABLE = hi;
                                       ! Enable Address Rus Ruffer
     EXABUF = IABUS;
                                       ! Gate Internal Address Bus
                                       ! Into External Address Buffer
```

北京 動名の内の名の内閣

```
FCMODE = SRMODE;
                                ! User Mode
FCSFACE = 2;
                                ! Accessing Program
                                ! Execute Pending Assignments
next;
ABUS = EXABUF;
                                ! Address flaced On Rus(Added)
                                ! Execute Pending Assignments
next;
T = 1;
                                ! Clock Cycle 1
                                ! Execute Assignment
next;
PHI1 = hi;
                                ! Phase 1 Of
                                ! Clock Cycle 1
FH12 = 10:
                                ! Assert Address Strobe
ASN = 10:
LISN = 10;
                                ! Assert Lower Data Strobe
                                ! Assert Upper Data Strobe
UDSN = 10;
IARUS = A[0];
                                ! Move Jump Address From A[O]
                                ! To Internal Address Buffer
DRENABLE = hi;
                                ! Enable Data Bus
                                ! Execute Pending Assignments
next;
                                ! Phase 2
FHI1 = 10;
                                ! Of Clock Cycle 1
FHI2 = hi;
PC = IABUS;
                                ! Place Jump Address Into Program
next;
7 = 2;
                                ! Clock Cycle 2
                                ! Execute Assignment
next;
                                ! Phase 1
PHI1 = hi;
                                ! Of Clock Cycle 2
PHI2 = 10;
while DTACKN eql hi
                               ! Wait For Memory To Place
                               ! Nata On The Rus
                                ! Execute Impending Assignments
     next;
                               ! Phase 2
     PHI1 = lo;
     PHI2 = hi;
                                ! Of Clock Cycle 2
                                ! Execute Assignments
     next;
     7 = 3;
                                ! Clock Cycle 3
     next;
                                ! Execute Assignment
     FHI1 = hi:
                                ! Phase 1
     PHI2 = 10;
                                ! Of Clock Cycle 3
                                ! Memory Places Instruction
     DBUS<15:8> = MEABUS];
     DBUS<7:0> = MCABUS + 13;
                                ! On Data Bus And
     IITACKN = 10;
                                ! Asserts ITACKN(Added)
     next;
                                ! Execute Pending Assignments
     ! Return To Phase 2
```

```
! Of Clock Cycle 2
     );
     next;
                                 ! Execute Impending Assignments
! Clock Cycle 3
next;
                                  ! Execute Assignment
F'HI1 = lo;
                                 ! Phase 2
                                  ! Of Clock Cycle 3
PHI2 = hi:
EXDBUF = DBUS;
                                  ! Instruction On Data Bus
                                 ! Is Placed In External Data
                                  ! Bus Buffer
                                 ! Execute Pending Assignments
next;
! Clock Cycle 4
T = 4;
                                  ! Execute Assignment
next;
                                 ! Phase 1
PHI1 = hi;
FHI2 = 10;
                                  ! Of Clock Cycle 4
next;
PFR = EXDBUF;
                                 ! The Contents Of The External
                                 ! Data Bus Buffer Are Placed
                                  ! In Prefetch Register
next;
                                  ! Execute Pending Assignments
                                  ! Phase 2
PHI1 = lo;
PHI2 = hi;
                                  ! Of Clock Cycle 4
ASN = hi;
                                  ! Deactivate Address Strobe
LDSN = hi;
                                 ! Deactivate Lower Nata Strobe
UDSN = hi;
                                  ! Neactivate Upper Data Strobe
DTACKN = hi;
                                  ! Deactivate Data Transfer
                                  ! Acknowledge(Added)
next;
T = 5;
                                 ! Clock Cycle 5
next:
                                 ! Execute Previous Assignment
                                  ! Phase 1 Of
PHI1 = hi:
PHI2 = 16;
                                  ! Clock Cycle 5
FiW = hii
                                  ! Memory Read
ADENABLE = 10;
                                  ! Disable Address Bus Ruffer
                                  ! Disable Data Bus Buffer
DBENABLE = 10;
IABUS = PC;
                                  ! Flace PC On Internal Address
                                  ! Rus
                                  ! Execute Pending Assignments
next;
                                 ! Phase 2 Of
PHI1 = 10;
PHI2 = hi;
                                  ! Clock Cycle 5
                                 ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2;
                                  ! Accessing Program
```

```
EXABUF = IABUS;
                                ! Gate Internal Address Bus
next;
                                ! Into External Address Buffer
ARUS = EXABUF;
                                ! Address Flaced On Rus(Added)
next;
                                ! Execute Pending Assignments
T = 6;
                                ! Clock Cycle 6
next;
                                ! Execute Assignment
                                ! Phase 1 Of
PHI1 = hi;
PHI2 = 16;
                                ! Clock Cycle 6
ASN = 10;
                                ! Assert Address Strobe
LDSN = lo;
                               ! Assert Lower Data Strobe
UDISN = lo;
                               ! Assert Upper Data Strobe
DBENABLE = hi;
                               ! Enable Data Rus
next;
                                ! Execute Pending Assignments
FHI1 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 6
next;
                                ! Execute Fending Assignments
T = 7;
                                ! Clock Cycle 7
next;
                                ! Execute Assignment
FHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 7
PHI2 = 10;
while ETACKN eql hi
                                ! Wait For Memory To Flace
                                ! Data On The Bus
    next;
                               ! Execute Impending Assignments
    FHI1 = 10;
                               ! Phase 2
    PHI2 = hi:
                                ! Of Clock Cycle 7
    next;
                               ! Execute Assignments
     ! Clock Cycle 8
    next;
                               ! Execute Assignment
                                ! Phase 1
    PHI1 = hi;
    PHI2 = 10;
                                ! Of Clock Cycle 8
                               ! Memory Places Instruction
    DBUS<15:8> = MEABUSJ;
    I(BUS<7:0> = MEARUS + 1);
                               ! On Data Bus And
                                ! Asserts DTACKN(Added)
    DTACKN = 10;
    next:
                                ! Execute Pending Assignments
     /*********************************
    T = 7
                               ! Return To Phase 2
                               ! Of Clock Cycle 7
     );
                               ! Execute Impending Assignments
     next;
```

```
T = 8;
                                          ! Clock Cycle 8
                                          ! Execute Assignment
     next;
     FHI1 = lo;
                                          ! Phase 2
                                          ! Of Clock Cycle 8
     PHI2 = hi;
     EXDBUF = DBUS;
                                          ! Instruction On Data Bus
                                          ! Is Flaced In External Data
                                          ! Rus Buffer
     next;
                                          ! Execute Pending Assignments
     T = 9:
                                          ! Clock Cycle 9
     next;
                                          ! Execute Assignment
     FHI1 = hi;
                                          1 Phase 1
     FHI2 = 10;
                                          ! Of Clock Cycle 9
     PFR = EXDBUF;
                                          ! The Contents Of The External
                                          ! Nata Bus Buffer Are Flaced
                                          ! In Prefetch Register
     next;
                                          ! Execute Fending Assignments
     PHI1 = 10;
                                          ! Phase 2
                                          ! Of Clock Cycle 9
     PHI2 = hi;
                                          ! Deactivate Address Strobe
     ASN = hi;
     LISN = hi;
                                          ! Neactivate Lower Nata Strobe
     UDSN = hi;
                                          ! Deactivate Upper Data Strobe
     PC = PC + 2;
                                          ! Increment Program Counter
     IR = PFR:
                                          ! Place Contents Of Prefetch
                                          ! Register Into Instruction
                                          ! Register
     IJTACKN = hi;
                                          ! Deactivate Data Transfer
                                          ! Acknowledge(Added)
                                          ! Execute Pending Assignments
     next;
                                          ! Reset Clock Cycle Counter
     T = 0
decode_execute_prefetch :=
                       case IR
                            0x3341: move
                                          ! MOVE.W [11,4(A1) [8(A1)]
                            0x027c: andi ! AND.W #$DFFF,SR
                            047320: Jmp
                                          ! JMF (AO) If IR = Octal Value
                       esac
                       )
main :=
    power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
          decode_execute_prefetch
```

```
11
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W D1,04(A1,D7) INSTRUCTION
                                                */
/*
                                                */
/x
                                                */
/*
              Structure Declarations
                                                */
/*
                                                */
state
/*
/*
           M68000 Programming Registers
                                                */
/*
                                                1/
DE0:73(31:0),
                   ! 8 Data Registers
A[0:63<31:0>,
                  ! 7 Address Registers
UA7<31:0>.
                   ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Pointer
PC<31:0>,
                   ! Program Counter
SR<15:0>,
                   ! Status Register
/x
                                                */
/*
           Temporary Internal Registers
                                                1/
/*
                                                x/
PFR<15:0>.
                   ! Prefetch Register
IR<15:0>,
                   ! Instruction Register
FC<2:0>,
                   ! Function Code Register
EXDBUF(15:0).
                   ! External Data Bus Buffer Register
EXABUF<23:1>.
                   ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>.
                   ! ALU Buffer 1
ALUBUF2<31:0>,
                   ! ALU Buffer 2
DTEMP<15:0>,
                   ! Temporary Data Storage
DISKEG<31:0>.
                   ! Temporary Displacement Storage
SRTEMP<15:0>,
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPADR(31:0>,
                   ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>,
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>,
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:0>,
                         ! Temporary Address Storage For
                         ! Exception Handler Coutine
T<7:0>,
                         ! Clock Cycle Counter
RESET,
                      ! Reset Flip-Flop
HALT,
                      ! Halt Flap-Flop
FiW,
                      ! Read/Write Flip-Flop
ADENABLE,
                        Address Rus Buffer Enable
DIBENABLE,
                      ! Data Bus Buffer Enable
ASN.
                      ! Address Strobe Flip-Flop
LIISN.
                      ! Lower Nata Strobe Flip-Flop
UIISN,
                      ! Upper Data Strobe Flip-Flop
ITACKII,
                      ! Nata Transfer Acknowledge Flip-Flop
COUT,
                      ! Carry Flip-Flop
EXCEPT.
                      ! Exception Processing Flip-Flop
REALIY.
                      ! Ready Flip-Flop
/*
                                                               */
/*
       Model transformation modifications:
                                                               ¥/
/×
                                                               */
/*
           1) CDL decoder structure nonexistent in ISP' and un-
                                                               */
14
       necessary for model. Eliminated.
                                                               */
/ x

    hulti-phase clock structure nonexistent in ISP*.

                                                               */
/*
       Operations on registers will provide its equivalent.
                                                               */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                               */
/*
       register will provide its equivalent.
                                                               */
/*

    The declared bus structures are modeled with registers */

/#
     . Without loss of model accurracy. This done to maintain model
                                                              */
14
       equivalency and simplicity.
                                                               K/
/*
           5) The memory word length was reduced from 16 to 8 bit
/±
       words to coincide with the ECR's 32-Kbyte memory, to agree with#/
/ #
       their FC incrementation, and to enable the use of existing
                                                              */
/*
       MC68000 assembler and linker/loader models. The memory was
                                                              1/
/ x
       ulso reduced from 8 Mwords to 32 Kbytes.
                                                              */
/ h
                                                              1/
IABUS(31:0),
                        ! Internal Address Bus
INBUS<31:0>,
                        ! Internal Data Rus
twait<4:0>,
                        ! Wait State Counter
SWITCH,
                      ! Fower Switch
PHI1,
                      ! Phase 1 Of Two-Phase Clock
                      ! Phase 2 Of Two-Phase Clock
FHI2;
port
/*
                                                              x/
/¥
             External Address and Data Bus
                                                              */
/*
                                                              */
DBUSK15:0>,
                        ! External Data Bus
```

```
ARUS (23:1);
                      ! External Address Bus(changed)
formut
*/
/*
                Register Subfields
                                                         */
                                                         */
/x
PCADDE
         = FC(23:0/)
                      ! Program Counter Address Field
SRTRACE
                      ! Trace Bit
         = SR<15>,
SRMODE
                      ! Mode Selection Bit
         = SR<13>,
SRCARRY
         = SR<0>,
                      ! Carry Bit
SKOVER
                      ! Overflow Bit
         = SR<1>,
SRZERO
                      ! Zero Bit
         = SR<2>,
SRNEG
                      ! Negative Bit
         = SR<3>,
SREX
         = SR<4>.
                      ! Extend Bit
SRMASK
         = SR<10:8>,
                      ! Interrupt Mask
FCSFACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCMODE
         ≈ FC<2>.
                      ! User/Supervisor Mode Bit
F'CLOW
         = PC<15:0>.
                      ! PC Low Word
PCHI
         = PC<31:16>,
                      ! PC High Word
DOLWORD
         = DE03<15:0>,
                      ! DICOJ Low Word
DILWORD
         = DC13<15:0>,
                      ! D[1] Low Word
D2LWORD
         = DC23<15:0>,
                      ! DE23 Low Word
I:3LWORD
         = DE33<15:0>.
                      ! D[3] Low Word
II4LWORD
         = 10043(15:0),
                      ! DI43 Low Word
DELWORD
         = DE53<15:0>,
                      ! DESI Low Word
II6LWORE
         = DL63<15:0>,
                      ! It63 Low Word
         = BE73<15:0>,
D7LWORD
                      ! DE73 Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
         = HANADR<15:0>, ! HANADR Low Word
HANATIRLOW
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADRLOW
         = TEMPATR<15:0>,! TEMPATR Low Word
TEMPADAHI
         = TEMPADR<31:16>;! TEMPADR High Word
Memory
11
                                                         */
                16K 16-Rit Word Internal Memory
                                                         */
/*
                                                         */
ME0:327673<7:0>;
mac no
/ x
                                                         */
/*
               Logic Level Macros
                                                         */
```

```
= 0 %,
10
    = 1 %,
hi
    = 0 %,
off
    = 1 %,
on
clear = 0 %;
/*
                                                        */
                                                        */
/* Power On and Initialization. This process was not modeled but is
                                                        11/
   added to initialize signals and registers.
/*
                                                        x/
power_on_initialize :=
      SWITCH = on;
                                ! Turn Power On
                                ! Execute Assignment
      next;
      READY = 10;
                                ! System Not Ready
                                ! Assert Reset For
      RESET = 10;
      delay(100);
                                ! 100 Miliseconds(Active Low)
      RESET = hi;
                                ! Deactivate Reset
      next;
                                ! Execute Fending Assignments
      ASN = hi:
                                ! Initialize Address Strobe
      LDSN = hi;
                                ! Initialize Lower Data Strobe
      UDSN = hi;
                                ! Initialize Upper Data Strobe
      TITACKN = hi;
                                ! Initialize Nata Transfer Acknowledge
                                ! Initialize Read/Write(Read On High)
      RW = hi;
                               ! Place Nata Bus In High Impedance State
      DBUS = 0xffff;
                                 ! Place Memory Locations Following The
      ME0 \times 100 e] = 0 \times ff;
      M[0x100f] = 0xff;
                                  ! JMF Instruction In A High State
                                ! Initialize Halt Flip-Flop(Active
      HALT = hi;
                                ! Low)
      T = 0:
                                ! Initialize Clock Cycle Counter
      READY = hi:
                                 ! System Ready
      /*
            Routine Initialization Per Hamby and Guillory
                                                        */
      /*
                                                        */
      ! Place Hex 5555 Into D[1]
      I([1]) = 0.5555;
                             ! Flace 6 Into D[7]
      I(7) = 0.00000006;
                                 ! Place Hex 1004 Into ACO3
      A[0] = 0 \times 1004;
                              ! Store Data At This Address
      A[1] = 0 \times 2000;
      FC = 0x1000;
                                ! Flace Hex 1000 Into Program Counter
      next
                                ! Execute Assignments
      )
Initial Fetch Cycle. This cycle was not modeled but is necessary
```

```
/* to retrieve modeled instructions for simulation and analysis. It
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
                                                           */
/* on page VI-15 of their thesis.
                                                           x/
/*
fetcn_initial_instruction :=
     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                     ! Clock Cycle 0
                                     ! Memory Read
     RW = hi;
     ADENABLE = 10:
                                     ! Disable Address Bus Buffer
                                     ! Disable Data Bus Buffer
     DRENABLE = 10;
     1ABUS = PC;
                                     ! Flace FC On Internal Address
                                     ! Bus
     next;
                                     ! Execute Pending Assignments
     PHI1 = lo:
                                     ! Phase 2 Of
     FHI2 = hi;
                                     ! Clock Cycle 0
                                     ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                     ! Gate Internal Address Rus
                                     ! Into External Address Buffer
     FCMODE = SRMODE:
                                     ! User Mode
     FCSPACE = 2;
                                     ! Accessing Program
                                     ! Execute Impending Assignments
     next:
                                    ! Address Placed On Rus(Added)
     ABUS = EXABUF;
     next;
                                     ! Execute Pending Assignments
     T = 1;
                                     ! Clock Cycle 1
     next;
                                     ! Execute Assignment
                                     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                     ! Clock Cycle 1
     ASN = 10;
                                     ! Assert Address Strobe
                                    ! Assert Lower Data Strobe
     LISN = lo:
                                    ! Assert Upper Data Strobe
     UIISN = lo:
     DBENABLE = hi;
                                    ! Enable Data Bus
     next;
                                     ! Execute Penaing Assignments
                                     ! Phase 2
     FHI1 = 16;
     PHI2 = hi;
                                     ! Of Clock Cycle 1
                                     ! Execute Pending Assignments
     next;
     7 = 2;
                                     ! Clock Cycle 2
                                     ! Execute Assignment
     next;
     PHI1 = hi;
                                     ! Phase 1
                                     ! Of Clock Cycle 2
     PHI2 = 10:
```

```
while DTACKN eql hi
                              ! Wait For Memory To Place
                              ! Data On The Bus
                              ! Execute Impending Assignments
    next;
                              ! Phase 2
    PHI1 = 10;
    PHI2 = hi:
                              ! Of Clock Cycle 2
    next:
                              ! Execute Assignments
     T = 3;
                               ! Clock Cycle 3
    next;
                               ! Execute Assignment
    PHI1 = hi;
                               ! Phase 1
                               ! Of Cluck Cycle 3
    PHI2 = 10;
    DBUS<15:8> = MEABUS];
                               ! Memory Places Instruction
                               ! On Data Bus And
     DBUS<7:0> = MEABUS + 13;
    DITACKN = 10;
                               ! Asserts DTACKN(Added)
    next:
                               ! Execute Pending Assignments
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
                              ! Execute Impending Assignments
    next:
! Clock Cycle 3
T = 3:
next;
                               ! Execute Assignment
F'HI1 = 10;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 3
                               ! Instruction On Data Rus
EXDRUF = DBUS;
                               ! Is flaced In External Data
                               ! kus Buffer
next;
                               ! Execute Pending Assignments
! Clock Cycle 4
T = 4;
                               ! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi;
                               ! Of Clock Cycle 4
FHI2 = 16;
                               ! The Contents Of The External
PFR = EXDRUF;
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
                               ! Execute Pending Assignments
next;
                               ! Phase 2
FH11 = 10;
PHI2 = h1;
                               ! Of Clock Cycle 4
                               ! Deactivate Address Strobe
ASN = hi:
                               ! Deactivate Lower Data Strobe
LISN = hi:
                               ! Deactivate Upper Data Strobe
UDSN = hi;
                               ! Contents Of Prefetch Register
IR = PFR;
```

```
! Are Placed Into Instruction
                                        ! Register
                                        ! Deactivate Data Transfer(Added)
     DTACKN = hi;
                                        ! Acknowledge
     PC = PC + 4;
                                        ! Increment Program Counter
                                        ! Execute Pending Assignments
     next;
                                        ! Reset Clock Cycle Counter
     T = 0
                                        ! AND W ##DFFF, SR
andi :=
    SRMODE = lo;
                                        ! Effect Of Instruction
                                        ! Prefetch Next Instruction
    IR<15:8> = MEPC3;
    IR\langle7:0\rangle = MEPC + 13;
                                        ! Is To Set Status Register
    next;
                                           ! Increment Program Counter
       PC = PC + 2;
    T = 5:
                                        ! Supervisor Bit To User
    next;
                                        ! Mode
    T = 0
                                        ! Requires 6 Clock Cycles
                                        ! MOVE.W D1,4(A1,D7) [B(A1,D7)]
810VE :=
     ! Phase 1 Of
     PHI1 = hi;
                                        ! Clock Cycle 0
     PHI2 = 10:
                                        ! Place Data Bus In High Impedance
     DRUS = Oxffff;
     RW = hi;
                                        ! Hemory Read
     ADENABLE = 10;
                                        ! Disable Address Bus Buffer
                                        ! Address Bus High Impedanced
     ARUS = 0xffffff;
     DRENABLE = 10;
                                        ! Disable Data Bus Buffer
     1ABUS<31:1> = PC<31:1>;
                                        ! Place PC On Internal Address
                                        ! Bus
     next;
                                        ! Execute Fending Assignments
                                        ! Phase 2 Of
     PHI1 = lo;
     PHI2 = hi:
                                        ! Clock Cycle 0
                                       ! Enable Address Bus Buffer
     ADENABLE = h1;
     EXABUF = IABUS<23:1>;
                                        ! Gate Internal Address Rus
                                       ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
                                        ! Accessing Program
     FCSFACE = 2;
     ABUS = IABUS<23:1>:
                                        ! Address Placed On Bus
     next;
                                        ! Execute Impending Assignments
     T = 1:
                                        ! Clock Cycle 1
     next;
                                        ! Execute Assignment
                                        ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                        ! Clock Cycle 1
```

```
ASN = 10:
                               ! Assert Address Strobe
LDSN = 10;
                               ! Assert Lower Data Strobe
                               ! Assert Upper Data Strobe
UBSN = lo;
                               ! Enable Data Rus
DEENABLE = hi;
                               ! Execute Pending Assignments
next;
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 1
PHI2 = hi;
                               ! Execute Pending Assignments
next:
T = 2:
                               ! Clock Cycle 2
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 2
while DTACKN eql hi
                               ! Wait For Memory To Flace
                              ! Data On The Rus
                              ! Execute Impending Assignments
    next;
    PHI1 = lo:
                              ! Phase 2
                               ! Of Cluck Cycle 2
     PHI2 = hi;
                               ! Execute Assignments
    next;
     T = 3;
                               ! Clock Cycle 3
    next;
                               ! Execute Assignment
     FHI1 = hi;
                               ! Phase 1
    PHI2 = 10;
                               ! Of Clock Cycle 3
    DRUS<15:8> = MEABUSD;
                               ! Memory Places Instruction
                               ! On Data Bus And
     DRUS<7:0> = MCABUS + 13;
                               ! Asserts DTACKN(Added)
    DITACKN = 16;
    next:
                               ! Execute Pending Assignments
     ! Return To Phase 2
     T = 2
                               ! Of Clock Cycle 2
    );
                              ! Execute Impending Assignments
     next;
! Clock Cycle 3
T = 3;
                               ! Execute Assignment
next;
FHI1 = 10:
                               ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
                               ! Instruction On Data Bus
EXUBUF = DBUS;
                               ! Is Placed In External Data
                               ! Bus Euffer
                               ! Execute Pending Assignments
next:
```

```
T = 4;
                                 ! Clock Eycle 4
next;
                                 ! Execute Assignment
                                ! Phase 1
PHI1 = hi;
FHI2 = 10:
                                ! Of Clock Cycle 4
DISREG = EXDBUF(7:0> sxt 32;
                                ! Store Displacement
                                ! Execute Pending Assignments
                                ! Phase 2
FHI1 = 10;
                                 ! Of Clock Cycle 4
f'HI2 = hi;
                                 ! Deactivate Address Strobe
ASN = hi;
                                 ! Neactivate Lower Nata Strobe
LDSN = hi;
                                 ! Deactivate Upper Data Strobe
UDSN = hi;
                                 ! Are Placed Into Instruction
                                 ! Register
                                 ! Increment Program Counter
PC = PC + 2;
INTACKN = hi:
                                ! Deactivate Data Transfer(Added)
                                ! Acknowledge
next;
! Clock Cycle 5
                                 ! Execute Previous Assignment
next;
                                ! Phase 1 Of
PHI1 = hi;
                                ! Clock Cycle 5
FH12 = 10;
                               ! Address Bus High Impedanced
ABUS = 0xffffff;
                                ! Nota Bus High Impedanced
DBUS = 0xffff;
DISREG = DISREG + AC11;
                             ! Add Address Register To Displacement
next;
                                ! Execute Fending Assignments
                                ! Phase 2 Of
PHI1 = 10;
PHI2 = hi:
                                 ! Clock Cycle 5
                                 ! Add Data Register To Displacement
DISREG = DISREG + D7LWORD;
                                 ! Into External Address Buffer
! Clock Cycle 6
T = 6;
                                 ! Execute Assignment
next;
FHI1 = hi:
                                 ! Phase 1 Of
                                 ! Clock Cycle 6
PH12 = 10;
next;
                                 ! Execute Pending Assignments
                                 ! Phase 2
PHI1 = 10;
PH12 = hi;
                                 ! Of Clock Cycle 6
                                 ! Execute Pending Assignments
next;
T = 7;
                                 ! Clock Cycle 7
                                 ! Execute Previous Assignment
next;
                                 ! Phase 1 Of
PHI1 = hi;
```

```
PHI2 = 10;
                                ! Clock Cycle 7
RW = hi;
                                ! Memory Read
ADENABLE = 10;
                                ! Disable Address Bus Buffer
DBENABLE = 10;
                                ! Disable Data Bus Buffer
IABUS<31:1> = PC<31:1>;
                                ! Place PC On Internal Address
next;
                                ! Execute Pending Assignments
                                ! Phase 2 Of
FHI1 = 10;
PHI2 = hi:
                                ! Clock Cycle 7
                                ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                               ! User Mode
                               ! Accessing Data
FCSFACE = 2;
EXABUF = IABUS(23:1);
                               ! Gate Internal Address Bus
ABUS = 1ABUS<23:1>;
                                ! Place Address On Rus
next:
                                ! Into External Address Buffer
T = 8;
                                ! Clock Cycle 8
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1 Of
FHI2 = 10;
                                ! Clock Cycle 8
                                ! Activate Upper And
UDSN = 10;
LDSN = 10;
                                ! Lower Data Strobes
ASN = 10;
                                ! Assert Address Strobe
                                ! Enable Data Bus
IBENABLE = hi:
next;
                                ! Execute Pending Assignments
FHII = 10;
                                ! Phase 2
PHI2 = h1;
                                ! Of Clock Cycle 8
                                ! Execute Pending Assignments
next;
! Clock Cycle 9
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1 Of
                                ! Clock Cycle 9
PHI2 = 10;
                                ! Wait For Memory To Place
while DTACKN eql hi
                                ! Data On The Bus
     next;
                                ! Execute Impending Assignments
     PHI1 = lo;
                                ! Phase 2
     FHI2 = hi;
                                ! Of Clock Cycle 9
                                ! Execute Assignments
     T = 10;
                                ! Clock Cycle 10
     next;
                                ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
     FHI2 = 10;
                                ! Of Clock Cycle 10
```

```
DBUS(15:8) = MCABUSJ;
                                ! Memory Places Instruction
    I(BUS(7:0) = MCABUS + 13;
                                ! On Data Rus And
    DTACKN = lo:
                                ! Asserts DTACKN(Added)
                                ! Execute Fending Assignments
    next;
     ! Return To Phase 2
                                ! Of Clock Cycle 9
     );
                                ! Execute Impending Assignments
    next;
! Clock Cycle 10
T = 10:
next;
                                ! Execute Assignment
PHI1 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 10
EXDRUF = DRUS;
                                ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Bus Buffer
                                ! Execute Pending Assignments
next;
T = 11;
                                 ! Clock Cycle 11
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 11
FHI2 = 10;
                                ! The Contents Of The External
PFR = EXDBUF;
                                ! Data Bus Buffer Are Flaced
                                ! In Prefetch Register
next;
                                ! Execute Pending Assignments
PHI1 = 10:
                                ! Phase 2
                                ! Of Clock Cycle 11
FHI2 = hi;
                                ! Deactivate Address Strobe
ASN = hi;
LDSN = hi:
                                ! Neactivate Lower Nata Strobe
                                ! Deactivate Upper Data Strobe
UDSN = hi;
                                ! Deactivate Data Transfer(Added)
IJTACKN = hi;
                                ! Acknowledge
                                ! Execute Pending Assignments
next;
T = 12;
                                 ! Clock Cycle 12
                                ! Execute Assignment
next;
                                ! Phase 1 Of
PHI1 = hi;
                                ! Clock Cycle 12
PHI2 = 10;
                                ! Place Data Bus In High Impedance
DIBUS = Oxffff;
                                ! Memory Read
RW = hi;
ADENABLE = 10;
                                ! Disable Address Bus Buffer
                                ! Address Bus High Impedanced
ABUS = 0xffffff;
                                ! Disable Data Bus Buffer
IIBENABLE = 10;
```

```
IABUS = DISKEG;
                                   ! Flace DISREG On Internal Address
                                   ! Bus
next;
                                   ! Execute Pending Assignments
FHI1 = 10;
                                  ! Phase 2 Of
PHI2 = hi;
                                  ! Clack Cycle 12
ADENABLE = hi:
                                  ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;
                                 ! Gate Internal Address Rus
                                  ! Into External Address Buffer
                                  ! Place Low Word Of DE13 On Rus
IDBUS = D1LWORD;
                                  ! User Mode
FCMODE = SRMODE:
FCSPACE = 1;
                                  ! Accessing Data
ABUS = IABUS<23:1>;
                                  ! Address Flaced On Rus
next:
                                   ! Execute Impending Assignments
T = 13;
                                   ! Clock Cycle 13
next;
                                   ! Execute Assignment
                                  ! Phase 1 Of
PHI1 = hi;
FHI2 = 10;
                                   ! Clock Cycle 13
ASN = lo;
                                   ! Assert Address Strobe
KW = 10;
EXDBUF = IDBUS;
                                  ! Place Contents Of Internal
                                  ! Data Bus Into External Data Buffer
                                   ! Reset Condition Code Bits
SRCARRY = 10;
SROVER = 10;
SKZERO = lo;
SANEG = 10;
                                  ! Execute Pending Assignments
next:
PHI1 = lo;
                                  ! Phase 2
                                  ! Of Clock Cycle 13
PHI2 = hi;
if EXDBUF eql 0
                                  ! Set Zero Condition Bit If Needed
  SRZERO = hi;
                                  ! Place Data On External Data Bus
DRUS = EXDRUF;
DBENABLE = hi;
                                   ! Enable Data Bus
next;
                                   ! Execute Pending Assignments
T = 14:
                                   ! Clock Cycle 14
                                   ! Execute Assignment
next:
PHI1 = hi;
                                  ! Phase 1
PHI2 = 10;
                                  ! Of Clock Cycle 14
if EXDBUF<15>
                                  ! Set Negative Condition Bit
  SRNEG = hi;
                                  ! If Needed
UDSN = 10;
                                  ! Activate Upper And
                                  ! Lower Data Strobes
LDSN = 10;
                                  ! Wait Cycle Counter Initialized
twait = 0;
next;
while DTACKN eql hi
                                  ! Wait For Memory To Place
                                  ! Data On The Bus
```

```
twait = twait + 1;
                                ! Increment Wait Cycle
     next;
                                ! Execute Impending Assignments
     PHI1 = 10;
                                ! Phase 2
     PHI2 = hi;
                                ! Of Clock Cycle 14
                                ! Execute Assignments
     next;
     ! Clock Cycle 15
     T = 15:
     next;
                                ! Execute Assignment
     PHI1 = hi:
                                ! Phase 1
     PHI2 = 10;
                                ! Of Clock Cycle 15
     if twait eql 2
                                ! hemory Responds After 2 Cycles
     M(ARUS) = DRUS<15:8>;
                                i Store Data From Bus
     MEABUS + 1J = IBUS<7:0>;
                                ! In Memory
                                ! Asserts DTACKN(Added)
     IJTACKN = 10
     ):
     next;
                                ! Execute Pending Assignments
     /************************************
     T = 14
                                ! Return To Phase 2
                                ! Of Clock Cycle 14
     );
     next;
                                ! Execute Impending Assignments
T = 15;
                                 ! Clock Cycle 15
next;
                                ! Execute Assignment
PHI1 = 10;
                                ! Phase 2
                                ! Of Clock Cycle 15
PHI2 = hi;
next;
                                ! Execute Pending Assignments
T = 16:
                                 ! Clock Cycle 16
next;
                                 ! Execute Assignment
PHI1 = hi:
                                ! Phase 1
PH12 = 16;
                                ! Of Clock Cycle 16
                                ! Execute Pending Assignments
next;
PHI1 = lo;
                                ! Phase 2
                                ! Of Clock Cycle 16
PHI2 = hi:
ASN = hi;
                                ! Deactivate Address Strobe
LDSN = hi;
                                ! Deactivate Lower Data Strobe
UDSN = hi;
                                ! Deactivate Upper Data Strobe
PC = PC + 2
                                ! Increment Program Counter
IR = PFR:
                                ! Flace Contents Of Prefetch
                                ! Register Into Instruction
                                ! Register
                                ! Deactivate Data Transfer
DIACKN = hi;
```

```
! Acknowledge(Added)
                                      ! Execute Pending Assignments
     next;
     1 = 0
                                      ! JMP (A0)
-: que.
     ! Phase 1 Of
     PHI1 = hi;
                                      ! Clock Cycle 0
     FHI2 = 16:
                                      ! Place Data Rus In A High Impedance
     IBUS = 0xffff;
     RW = hi;
                                      ! Memory Read
     ADENABLE = 10:
                                      ! Disable Address Bus Buffer
     IIBENABLE = 10;
                                      ! Disable Data Bus Buffer
                                      ! Place PC On Internal Address
     IABUS = PC;
                                      ! Rus
                                      ! Execute Pending Assignments
     next;
     FHI1 = lo;
                                      ! Phase 2 Of
     PHI2 = hi;
                                      ! Clock Cycle O
     ADENABLE = hi;
                                      ! Enable Address Rus Buffer
     EXABUF = IABUS;
                                      ! Gate Internal Address Bus
                                      ! Into External Address Buffer
     FCMODE = SRMODE;
                                      ! User Mode
     FCSFACE = 2;
                                      ! Accessing Program
                                      ! Execute Pending Assignments
     next;
                                      ! Address Flaced Un Rus(Added)
     ABUS = EXABUF;
                                      ! Execute Pending Assignments
     next:
     7 = 1;
                                      ! Clock Cycle 1
     next:
                                      ! Execute Assignment
                                      ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                      ! Clock Cycle 1
     ASN = lo;
                                      ! Assert Address Strobe
                                      ! Assert Lower Data Strobe
     LDSN = 10;
     UDSN = lo;
                                      ! Assert Upper Data Strobe
                                      ! Move Jump Address From A[0]
     IABUS = A[0];
                                      ! To Internal Address Buffer
                                      ! Enable Data Rus
     DBENABLE = hi;
                                      ! Execute Pending Assignments
     next;
                                      ! Phase 2
     PHI1 = lo;
                                      ! Of Clock Cycle 1
     FHI2 = hi;
     PC = IABUS;
                                      ! Place Jump Address Into Program
                                      ! Counter
     next;
```

! Clock Cycle 2

T = 2:

```
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
FHI2 = 10;
while DTACKN eql hi
                               ! Wait For Memory To Flace
                               ! Data On The Rus
                               ! Execute Impending Assignments
    next;
    PHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 2
    PHI2 = hi;
    next;
                               ! Execute Assignments
     T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
    next:
    PHI1 = hi;
                               ! Phase 1
    PHI2 = 10;
                               ! Of Clock Cycle 3
    DBUS<15:8> = MEABUS3;
                               ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
    DITACKN = 10;
                               ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
    next;
     ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
    next;
                               ! Execute Impending Assignments
T = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
                               ! Phase 2
PHI1 = lo;
PHI2 = hi;
                               ! Of Clock Cycle 3
EXDRUF = DBUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 4
next;
FFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
```

```
! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                                 ! Deactivate Address Strobe
LDSN = hi;
                                 ! Deactivate Lower Data Strobe
UDSN = hi;
                                 ! Deactivate Upper Data Strobe
DTACKN = hi;
                                 ! Deactivate Data Transfer
                                  ! Acknowledge(Added)
next;
T = 5;
                                  ! Clock Cycle 5
                                  ! Execute Previous Assignment
next;
                                 ! Phase 1 Of
PHI1 = hi;
FHI2 = 10;
                                 ! Clock Cycle 5
RW = hi;
                                 ! Memory Read
                                  ! Disable Address Rus Ruffer
ADENABLE = 10;
DBENABLE = 10;
                                  ! Disable Data Bus Buffer
IABUS = FC;
                                  ! Place PC On Internal Address
                                  I Rus
                                  ! Execute Pending Assignments
next;
                                 ! Phase 2 Of
FHI1 = 10;
                                 ! Clock Cycle 5
PHI2 = hi;
ADENABLE = h:;
                                 ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2;
                                 ! Accessing Program
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
                                 ! Into External Address Ruffer
next;
ABUS = EXABUF;
                                 ! Address Placed On Bus(Added)
next;
                                 ! Execute Pending Assignments
! Clock Cycle 6
T = 6;
                                  ! Execute Assignment
next;
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 6
ASN = lo;
                                 ! Assert Address Strobe
                                 ! Assert Lower Data Strobe
LISN = lo;
                                 ! Assert Upper Nata Strobe
UDSN = lo:
                                 ! Enable Data Bus
LIBENABLE = hi;
                                 ! Execute Pending Assignments
next;
FHII = lo:
                                  ! Phuse 2
PHI2 = hi;
                                  ! Of Clock Cycle 6
next;
                                 ! Execute Fending Assignments
1 = 7;
                                 ! Clock Cycle 7
                                 ! Execute Assignment
next;
                                 ! Phase 1
FHI1 = hi;
                                 ! Of Clock Cycle 7
PHI2 = 10:
                                 ! Wait For Memory To Place
while DTACKN eql hi
```

いいという国際アンスティスと

```
! Data On The Bus
     next;
                                ! Execute Impending Assignments
                               1 Phase 2
     PHI1 = lo;
     PHI2 = hi;
                               ! Of Clack Cycle 7
                               ! Execute Assignments
     next;
     ! Clock Cycle 8
                               ! Execute Assignment
     next;
     PHI1 ≈ hi;
                                ! Phase 1
                                ! Of Clock Cycle 8
     fH12 = 10;
     DBUS<15:8> = MEABUS];
                               ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
                                ! Asserts DTACKN(Added)
     DITACKN = 10;
                               ! Execute Pending Assignments
     next;
     \************************\
                              ! Return To Phase 2
                               ! Of Clock Cycle 7
     );
                               ! Execute Impending Assignments
     next;
T = 8:
                                ! Clock Cycle 8
next;
                                ! Execute Assignment
FHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 8
EXDBUF = DBUS;
                               ! Instruction On Nata Bus
                               ! Is flaced In External Data
                               ! Rus Ruffer
next;
                                ! Execute Pending Assignments
T = 9;
                                ! Clock Cycle 9
                                ! Execute Assignment
next;
PHI1 = hi;
                                ! Phase 1
FHI2 = 10;
                                ! Of Clock Cycle 9
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                                ! In Prefetch Register
                                ! Execute Pending Assignments
next;
                                ! Phase 2
FHI1 = lo;
PHI2 = hi;
                                ! Of Clock Cycle 9
ASN = h1;
                                ! Deactivate Address Strobe
                               ! Deactivate Lower Data Strobe
LDSN = hi:
UDSN = hi;
                               ! Deactivate Upper Data Strobe
PC = PC + 2;
                               ! Increment Program Counter
IR = PFR;
                                ! Place Contents Of Frefetch
```

```
! Register Into Instruction
                                             ! Register
      DTACKN = hi;
                                             ! Deactivate Data Transfer
                                             ! Acknowledge(Added)
      next;
                                             ! Execute Pending Assignments
      T = 0
                                             ! Reset Clock Cycle Counter
decode_execute_prefetch !=
                        case IR
                             0x3381: move
                                             ! MOVE.W D1,4(A1,D7) [8(A1,D7)]
                             0x027c: andi
                                            ! AND.W #$DFFF,SR
                                             ! JMF (AO) If IR = Octal Value
                             047320: двр
                        esac
                        )
main :=
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
           decode_execute_prefetch
```

```
/*
                                                  ¥/
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W D1,$2004 INSTRUCTION
                                                  */
/*
                                                  1/
/*
                                                  */
11
              Structure Declarations
                                                  1/
/x
                                                  */
state
/*
                                                  */
/*
                                                  */
           M68000 Programming Registers
/*
                                                  x/
DE0:73<31:0>,
                   ! 8 Data Registers
                   ! 7 Address Registers
AE0:63<31:0>,
UA7<31:0>,
                   ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Pointer
PC<31:0>,
                   ! Program Counter
SR<15:0>,
                   ! Status Register
/*
                                                  */
            Temporary Internal Registers
                                                  */
/*
/*
                                                  */
PFR<15:0>.
                   ! Frefetch Register
IR<15:0>,
                   ! Instruction Register
FC<2:0>,
                   ! Function Code Register
EXDBUF<15:0>.
                   ! External Data Bus Buffer Register
EXABUF<23:1>.
                   ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>.
                   ! ALU Buffer 1
ALUBUF2<31:0>.
                   ! ALU Ruffer 2
DIEMPK15:0>.
                   ! Temporary Data Storage
DISREG<31:0>,
                   ! Temporary Displacement Storage
SRTEMP<15:0>,
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
                   ! Temporary Instruction Register Storage
IRTEMP<15:0>,
                   ! (Exception Processing)
                   ! Temporary Cycle Address Storage
TEMPAUR<31:0>.
                   ! (Exception Processing)
ACTYPE<15:0>.
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>,
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:0>,
                         ! Temporary Address Storage For
                         ! Exception Handler Routine
T<7:0>,
                         ! Clock Cycle Counter
                      ! Reset Flip-Flop
RESET,
HALT,
                      ! Halt Flip-Flop
R₩,
                      ! Read/Write Flip-Flop
                      ! Address Bus Buffer Enable
ADENABLE,
                      ! Nata Bus Buffer Enable
DIBENABLE,
                      ! Address Strobe Flip-Flop
ASN.
LIISN.
                      ! Lower Nata Strobe Flip-Flop
                      ! Upper Data Strobe Flip-Flop
UDISN.
DITACKN.
                      ! Note Transfer Acknowledge Flip-Flop
COUT,
                      ! Carry Flip-Flop
                      ! Exception Processing Flip-Flop
EXCEPT.
READY,
                      ! Ready Flip-Flop
/*
                                                                 */
       model transformation modifications:
/*
                                                                */
/x
           1) CDL decoder structure nonexistent in ISP' and un-
                                                                 */
/*
       necessary for model. Eliminated.
                                                                 x/
/x
           2) Multi-phase clock structure negexistent in ISP'.
                                                                 */
/×
       Operations on registers will provide its equivalent.
                                                                */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                                */
/*
       register will provide its equivalent.
                                                                x/
            4) The declared bus structures are modeled with registers */
/*
       without loss of model accurracy. This done to maintain model
                                                                 */
/*
/×
       equivalency and simplicity.
                                                                 */
                                                                 */
/*
            5) The memory ward length was reduced from 16 to 8 bit
/*
       words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*
       their PC incrementation, and to enable the use of existing
                                                                 */
/*
       MC68000 assembler and linker/loader models. The memory was
                                                                 1/
       also reduced from 8 Mwords to 32 Kbytes.
                                                                 1/
/ ¥
                                                                 */
/*
IARUS<31:0>,
                         ! Internal Address Bus
IDBUS<31:0>,
                         ! Internal Data Bus
                         ! Wait State Counter
twait<4:0>,
SWITCH,
                      ! Power Switch
PHI1,
                      ! Phase 1 Of Two-Phase Clock
                       ! Phase 2 Of Two-Phase Clock
PHI2:
port
                                                                1/
/1
                                                                 */
/*
              External Address and Data Bus
                                                                */
DBUS<15:0>,
                         ! External Data Bus
```

```
ABUS<23:1>;
                      ! External Address Bus(changed)
format
/*
                                                        */
/*
                Register Subfields
                                                        */
11
                                                        */
PCAUDR
         = PC<23:0>,
                      ! Program Counter Address Field
                      ! Trace Bit
SRTRACE
         = SR<15>.
                      ! Mode Selection Bit
SRHODE
         = SR<13>
                      ! Carry Bit
SRCARRY
         = SR<0>,
SROVER
                      ! Overflow Bit
         = SR<1>,
                      ! Zero Bit
SRZERO
         = SR<2>,
SKNEG
         = SR<3>,
                      ! Negative Bit
SREX
         = SR<4>,
                      ! Extend Bit
SRMASK
         = SR<10:8>,
                      ! Interrupt Mask
FCSFACE
         = FC<1:0>,
                      ! Memory Access Address Space
                      ! User/Supervisor Mode Bit
FCMODE
         = FC<2>,
F'CLOW
         = PC<15:0>.
                      ! FC Low Word
                      ! PC High Word
PCHI
         = PC(31116),
DOLWORD
         = DE03<15:0>,
                      ! DEOJ Low Word
DILWORD
         = DC13<15:0>,
                      ! DE13 Low Word
II2LWORD
         = DC23<15:0>,
                      ! II[2] Low Word
                      ! DE33 Low Word
D3LWORD
         = DC33<15:0>,
II4LWORE
         = D[43<15:0>,
                      ! DE43 Low Word
DSLWORD
                      ! D[5] Low Word
         = DE53<15:0>,
LIGLWORD
         = B[6](15:0).
                      ! DE63 Low Word
D7LWORD
         = DE73<15:0>.
                     ! DE73 Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANATIRLOW
         = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPAURLOW = TEMPAUR(15:0>,! TEMPAUR Low Word
         = TEMPADR<31:16>;! TEMPADR High Word
TEMPADRHI
DIGMOTY
/ k
                                                        */
/*
                16K 16-Bit Word Internal Memory
                                                        */
/*
                                                        1/
MEO:327673<7:0>;
DIGETO
/x
/x
               Logic Level Macros
                                                        1/
```

```
10
hi
    = 1 %.
off
    = 0 %,
OΠ
    = 1 %.
clear = 0 %;
/*
                                                          x/
   Fower On and Initialization. This process was not modeled but is
                                                          */
/*
   added to initialize signals and registers.
                                                          */
/*
                                                          */
power on initialize :=
                                  ! Turn Power On
      SWITCH = on;
                                  ! Execute Assignment
      next:
      READY = lo:
                                  ! System Not Ready
      RESET = 10;
                                 ! Assert Reset For
                                  ! 100 Miliseconds(Active Low)
      delay(100);
      RESET = hi;
                                 ! Deactivate Reset
                                  ! Execute Pending Assignments
      next;
      ASN = hi;
                                 ! Initialize Address Strobe
      LUSN = hi;
                                  ! Initialize Lower Data Strobe
      UDSN = hi;
                                 ! Initialize Upper Data Strobe
                                  ! Initialize Data Transfer Acknowledge
      DIACKN = hi;
                                 ! Initialize Read/Write(Read On High)
      8W = hi;
                                 ! Place Data Bus In High Impedance State
      DBUS = Oxffff;
                                  ! Place Memory Locations Following The
      M[0x100e] = 0xff;
      ME0::100f3 = 0:ff;
                                   ! JMP Instruction In A High State
      HALT = hi;
                                  ! Initialize Halt Flip-Flop(Active
                                  ! Low)
                                  ! Initialize Clock Cycle Counter
      T = 0;
                                  ! System Ready.
      READY = hi;
      */
      /*
            Routine Initialization Per Hamby and Guillory
                                                          */
      /×
      /x
                                                          */
      I([1] = 0.5555;
                              ! Place Hex 5555 Into DC1]
      A[0] = 0x1004;
                                  ! Place Hex 1004 Into A[0]
                                  ! Place Hex 1000 Into Program Counter
      FC = 0 \times 1000;
                                  ! Execute Assignments
      next
      )
/*
/x
   Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                          */
   to retrieve modeled instructions for simulation and analysis. It
   was fashsioned from the Read Cycle described by Hamby and Guillory */
```

```
*/
/# on page VI-15 of their thesis.
                                                           */
fetch_initial_instruction :=
     ! Phase 1 Of
     PHI1 = hi;
    PHI2 = 10;
                                     ! Clock Cycle 0
     RW = hi;
                                     ! Memory Read
     ADENABLE = 10;
                                     ! Disable Address Bus Buffer
     DRENABLE = 10;
                                    ! Disable Data Bus Buffer
                                    ! Fluce PC On Internal Address
     TABUS = PC:
                                    ! Bus
                                    ! Execute Fending Assignments
    next;
    PHI1 = 10;
                                    ! Phase 2 Of
    PHI2 = hi;
                                    ! Clock Cycle 0
    ADENABLE = hi;
                                    ! Enable Address Bus Ruffer
    EXABUF = IABUS;
                                    ! Gate Internal Address Rus
                                    ! Into External Address Buffer
                                    ! User Mode
     FCMODE = SRMODE;
    FCSFACE = 2;
                                    ! Accessing Program
    next;
                                    ! Execute Impending Assignments
     ABUS = EXABUF;
                                    ! Address Placed On Rus(Added)
                                    ! Execute Pending Assignments
     next;
     T = 1;
                                    ! Clock Cycle 1
     next;
                                    ! Execute Assignment
     PHI1 = hi;
                                    ! Phase 1 Of
                                    ! Clock Cycle 1
     PHI2 = lo;
                                    .! Assert Address Strobe
     ASN = lo;
                                    ! Assert Lower Data Strobe
     LIISN = 10;
                                    ! Assert Upper Data Strobe
     UDSN = 10;
     DIBENABLE = hi;
                                    ! Enable Data Rus
     next;
                                    ! Execute Pending Assignments
     PHI1 = 10;
                                    ! Phase 2
                                    ! Of Clock Cycle 1
     PHI2 = hi;
                                    ! Execute Fending Assignments
     next;
     T = 2;
                                    ! Clock Cycle 2
                                    ! Execute Assignment
     next;
     FHI1 = hi;
                                    ! Phase 1
                                    ! Of Clock Cycle 2
     PHI2 = 10;
                                    ! Wait For Memory To Place
     while DTACKN eql hi
                                    ! Data On The Bus
```

```
next;
                                ! Execute Impending Assignments
     PHI1 = 10;
                                ! Phase 2
     PHI2 = hi;
                                ! Of Clock Cycle 2
     next:
                                ! Execute Assignments
     T = 3;
                                ! Clock Cycle 3
     next;
                                ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
     FHI2 = 10:
                                ! Of Clock Cycle 3
     DBUS<15:8> = MLABUSJ;
                                ! Memory Places Instruction
                               ! On Data Bus And
     DBUS<7:0> = MCABUS + 1];
     DITACKN = 10;
                                ! Asserts DTACKN(Added)
     next;
                                ! Execute Fending Assignments
     T = 2
                                ! Return To Phase 2
                                ! Of Clock Cycle 2
     );
     next;
                                ! Execute Impending Assignments
7 = 3:
                                ! Clock Cycle 3
next;
                                ! Execute Assignment
FHI1 = 10;
                                ! Phase 2
PH12 = hi;
                                ! Of Clock Cycle 3
EXDBUF = DBUS;
                                ! Instruction On Data Rus
                                ! Is Placed In External Data
                                ! Bus Buffer
next;
                                ! Execute Pending Assignments
T = 4;
                                ! Clock Cycle 4
next;
                                ! Execute Assignment
FHI1 = hi;
                                ! Phuse 1
PHI2 = 10;
                                ! Of Clock Cycle 4
                                ! The Contents Of The External
PFR = EXDBUF;
                                ! Data Bus Buffer Are Placed
                                ! In Prefetch Register
next;
                                ! Execute Pending Assignments
FHI1 = 10;
                               ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 4
ASN = hi:
                                ! Deactivate Address Strobe
LISN = h1:
                                ! Deactivate Lower Data Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
IR = PFR:
                                ! Contents Of Prefetch Register
                                ! Are Placed Into Instruction
                                ! Register
```

```
! Deactivate Data Transfer(Added)
     ITACKN = hi;
                                        ! Acknowledge
                                        ! Increment Program Counter
     PC = PC + 4;
                                       ! Execute Pending Assignments
     next;
                                        ! Reset Clock Cycle Counter
     T = 0
                                        ! AND.W #$DFFF,SR
andi :=
    SRMODE = lo;
                                       ! Effect Of Instruction
                                        ! Prefetch Next Instruction
    IR<15:8> = MCPCJ;
    IR<7:0> = MEPC + 13;
                                        ! Is To Set Status Register
    next;
       PC = PC + 2;
                                          ! Increment Program Counter
                                        ! Supervisor Bit To User
    T = 5;
                                        ! Mode
    next;
                                        ! Requires 6 Clock Cycles
    T = 0
                                       ! MOVE.W D1,$2004 [2008]
move :=
     ! Phase 1 Of
     PHI1 = hi;
                                       ! Clock Cycle 0
     PHI2 = 10;
                                       ! Place Data Bus In High Impedance
     DBUS = 0xffff;
                                       ! Memory Read
     RW = hi:
                                       ! Disable Address Bus Buffer
     ADENABLE = lo;
                                       ! Address Bus High Impedanced
     ABUS = 0xffffff;
     DRENABLE = 10;
                                       ! Disable Data Bus Buffer
                                       ! Place PC On Internal Address
     IABUS<31:1> = PC<31:1>;
                                       ! Bus
                                        ! Execute Pending Assignments
     next;
     PHI1 = lo;
                                       ! Phase 2 Of
     PHI2 = hi;
                                       ! Clock Cycle 0
                                       ! Enable Address Bus Buffer
     ADENABLE = hi;
                                       ! Gate Internal Address Bus
     EXABUF = IABUS<23:1>;
                                      ! Into External Address Buffer
     FCMODE = SRMODE;
                                       ! User Mode
     FCSFACE = 2;
                                       ! Accessing Program
                                       ! Address Placed On Bus
     ABUS = IABUS<23:1>;
                                       ! Execute Impending Assignments
     next;
     ! Clock Cycle 1
     T = 1;
     next;
                                       ! Execute Assignment
     PHI1 = hi;
                                       ! Phase 1 Of
                                       ! Clock Cycle 1
     PHI2 = 10;
                                       ! Assert Address Strobe
     ASN = lo;
     LDSN = lo;
                                       ! Assert Lower Data Strobe
```

```
UIISN = lo:
                             ! Assert Upper Data Strobe
                             ! Enable Data Bus
DBENABLE = hi;
next:
                             ! Execute Fending Assignments
PHI1 = 10:
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 1
                             ! Execute Pending Assignments
next;
T = 2;
                             ! Clock Cycle 2
next;
                             ! Execute Assignment
PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 2
PHI2 = 10;
                             ! Wait For Memory To Place
while DTACKN eql hi
                             ! Data On The Bus
    next;
                             ! Execute Impending Assignments
    PHI1 = 10
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 2
    next;
                             ! Execute Assignments
    T = 3:
                             ! Clock Cycle 3
    next;
                             ! Execute Assignment
    PHI1 = hi:
                             ! Phase 1
                             ! Of Clock Cycle 3
    PHI2 = 10;
    DBUS<15:8> = MEABUSI;
                             ! Memory Places Instruction
    DRUS<7:0> = HEABUS + 13;
                             ! On Data Bus And
                             ! Asserts DTACKN(Added)
    TITACKN = 10;
    next;
                             ! Execute Pending Assignments
    T = 2
                             ! Return To Phase 2
                             ! Of Clock Cycle 2
    );
                            ! Execute Impending Assignments
    next;
! Clock Cycle 3
T = 3;
                             ! Execute Assignment
next;
FHI1 = 10:
                             ! Phase 2
                             ! Of Clock Cycle 3
PHI2 = hi:
EXDBUF = DBUS;
                             ! Instruction On Data Bus
                             ! Is Placed In External Data
                             ! Bus Buffer
next;
                             ! Execute Pending Assignments
T = 4;
                             ! Clock Cycle 4
                             ! Execute Assignment
next;
```

```
! Phase 1
PHI1 = hi;
                                   ! Of Clock Cycle 4
PH12 = 10;
DISREG = EXDBUF sxt 32;
                                   ! Store Displacement
next;
                                   ! Execute Pending Assignments
                                   ! Phase 2
PHI1 = lo:
PHI2 = hit
                                   ! Of Clock Cycle 4
ASN = hi;
                                   ! Deactivate Address Strobe
LISN = hi:
                                   ! Deactivate Lower Data Strobe
UDSN = hi;
                                   ! Deactivate Upper Data Strobe.
                                   ! Are Placed Into Instruction
                                   ! Register
PC = PC + 2;
                                   ! Increment Program Counter
DITACKN = hi;
                                   ! Deactivate Data Transfer(Added)
                                   ! Acknowledge
next;
T = 5;
                                   ! Clock Cycle 5
next:
                                   ! Execute Previous Assignment
PHI1 = h1;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 5
                                   ! Memory Read
RW = hi;
ADENABLE = 10;
                                   ! Disable Address Bus Buffer
ABUS = 0xffffff;
                                  ! Address Bus High Impedanced
                                   ! Data Bus Returned To High
DBUS = Oxffff;
                                   ! Impedance State
DEENABLE = 10;
                                   ! Disable Data Bus Buffer
                                   ! Place PC On Internal Address
TABUS<31:1> = PC<31:1>;
                                   ! Rus
next;
                                    Execute Pending Assignments
                                   ! Phase 2 Of
PHI1 = lo;
                                   ! Clock Cycle 5
PHI2 = hi;
                                   ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                   ! User Mode
FCSPACE = 2;
                                  ! Accessing Data
EXABUF = IABUS<23:1>:
                                  ! Gate Internal Address Bus
ABUS = IABUS<23:1>;
                                  ! Place Address On Bus
                                   ! Into External Address Buffer
T = 6;
                                   ! Clock Cycle 6
next;
                                   ! Execute Assignment
PHI1 = hi;
                                   ! Phase 1 Of
                                   ! Clock Cycle 6
PHI2 = lo;
UDSN = lo;
                                   ! Activate Upper And
LUSN = lo:
                                   ! Lower Data Strobes
ASN = 10;
                                   ! Assert Address Strobe
DBENABLE = hi;
                                   ! Enable Data Bus
```

```
nexti
                             ! Execute Fending Assignments
PHI1 = lo;
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 6
                             ! Execute Pending Assignments
nesti
T = 7;
                             ! Clock Cycle 7
next;
                             ! Execute Assignment
FRII = hi;
                             ! Phase 1 Of
f(H12 = 16)
                             ! Clock Cycle 7
while DTACKN eqt hi
                             ! Wait For hemory To Place
                             ! Lata On The Bus
    next;
                             ! Execute Impending Assignments
    FHI1 = 10:
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 7
    next;
                             ! Execute Assignments
    ! Clock Cycle 8
    next;
                             ! Execute Assignment
    PHI1 = bi:
                             ! Phase 1
    PHI2 = 10:
                             ! Of Clock Cycle &
    DRUS (15:8) = MEABUSH;
                             ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                             ! On Nate Bus And
    DITACKN = Ju;
                             ! Asserts DTACKN(Added)
    nexti
                             ! Execute Pending Assignments
    T = 7
                             ! Return To Phase 2
                             ! Of Clock Cycle 7
    );
                            ! Execute Impending Assignments
    next;
T = 8:
                             ! Clock Cycle 8
next;
                             ! Execute Assignment
PHII = lo;
                             ! Phase 2
                             ! Of Clock Cycle 3
PHI2 = hi;
EXDUUF = DRUS;
                             ! Instruction On Data Bus
                             ! Is Placed In External Data
                             ! bus Buffer
                             ! Execute Pending Assignments
next;
! Clock Cycle 9
T = 9;
next:
                             ! Execute Assignment
PHI1 = hi;
                             ! Phase 1
```

```
! Of Clock Cycle 9
PHI2 = 10;
                                   ! The Contents Of The External
PFR = EXDBUF;
                                    ! Data Bus Buffer Are Placed
                                   ! In Prefetch Register
                                    ! Execute Pending Assignments
next;
                                   ! Phase 2
FHI1 = lo:
                                    ! Of Clock Cycle 9
PHI2 = hi;
                                   ! Deactivate Address Strobe
ASN = hi:
                                   ! Neactivate Lower Nata Strobe
LDSN = hi:
                                   ! Deactivate Upper Data Strobe
UDSN = hi;
                                    ! Deactivate Data Transfer(Added)
ITACKN = hi;
                                    ! Acknowledge
                                    ! Execute Pending Assignments
next;
! Clock Cycle 10
T = 10;
                                    ! Execute Assignment
next;
                                    ! Phase 1 Of
PHI1 = hi:
PHI2 = 10;
                                    ! Clock Cycle 10
                                    ! Place Nata Rus In High Impedance
DBUS = 0xffff;
RW = hi;
                                    ! Memory Read
ADENABLE = 10;
                                    ! Disable Address Bus Buffer
                                    ! Address Bus High Impedanced
ABUS = 0xffffff;
                                    ! Disable Data Bus Buffer
IIBENABLE = 10;
                                    ! Place DISREG On Internal Address
IABUS = DISREG:
                                    ! Execute Pending Assignments
next;
                                    ! Phase 2 Of
PHI1 = lo;
                                    ! Clock Cycle 10
PHI2 = hi;
                                    ! Enable Address Bus Buffer
ADENABLE = hi;
                                    ! Gate Internal Address Bus
EXABUF = IABUS<23:1>;
                                    ! Into External Address Buffer
                                    ! Place Low Word Of D[1] On Bus
IDBUS = DILWORD;
                                    ! User Mode
FCMODE = SRMODE:
FCSPACE = 1;
                                    ! Accessing Data
                                    ! Address Placed On Rus
ARUS = IARUS<23:1>;
                                    ! Execute Impending Assignments
next:
T = 11;
                                    ! Clock Cycle 11
                                   .! Execute Assignment
next;
PHI1 = hi;
                                    ! Phase 1 Of
                                    ! Clock Cycle 11
FHI2 = 10;
ASN = lo;
                                    ! Assert Address Strobe
RW = 10;
EXDRUF = IDBUS;
                                    ! Flace Contents Of Internal
                                    ! Data Bus Into External Data Buffer
SRCARRY = lo;
                                    ! Reset Condition Code Bits
SROVER = 10;
```

```
SRZERO = lo;
SRNEG = 10;
                                 Execute Pending Assignments
next;
                                 ! Phase 2
PHI1 = 10;
                                 ! Of Clock Cycle 11
PHI2 = hi;
                                 ! Set Zero Condition Bit If Needed
if EXDRUF eq1 0
  SRZERO = hi;
DRUS = EXPRUF:
                                 ! Place Data On External Data Bus
                                 ! Enable Data Bus
DRENABLE = hi;
                                 ! Execute Pending Assignments
next;
T = 12;
                                  ! Clock Cycle 12
next;
                                 ! Execute Assignment
PHI1 = hi:
                                 ! Phose 1
                                 ! Of Clock Cycle 12
PHI2 = 10;
                                 ! Set Negative Condition Bit
if EXDBUF<15>
                                 ! If Needed
  SRNEG = hi;
UDSN = 10:
                                 ! Activate Upper And
                                 ! Lower Data Strobes
LDSN = lo;
                                 ! Wait Cycle Counter Initialized
twait = 0;
next;
                                 ! Wait For Memory To Place
while DTACKN eql hi
                                 ! Data On The Bus
     twait = twait + 1;
                                 ! Increment Wait Cycle
     next;
                                 1 Execute Impending Assignments
     F'HI1 = 10;
                                 ! Phase 2
     PHI2 = hi;
                                 ! Of Clock Cycle 12
     next;
                                 ! Execute Assignments
     T = 13;
                                 ! Clock Cycle 13
     next;
                                 ! Execute Assignment
                                 ! Phase 1
     FHI1 = hi;
                                 ! Of Clock Cycle 13
     PHI2 = 10;
                                 ! Memory Responds After 2 Cycles
     if twait eql 2
     MCABUS] = DBUS(15:8);
                                 ! Store Nata From Rus
     MEABUS + 13 = DBUS<7:0>;
                                 ! In Memory
                                ! Asserts DTACKN(Added)
     ITACKN = 10
     ):
                                 ! Execute Pending Assignments
     next;
     ! Return To Phase 2
     T = 12
                                ! Of Clock Cycle 12
     );
                               ! Execute Impending Assignments
     next;
```

```
7 = 13;
                                       ! Clock Cycle 13
                                      ! Execute Assignment
     next;
     FH11 = lo;
                                      ! Phase 2
                                      ! Of Clock Cycle 13
     PHI2 = hi;
     next:
                                      ! Execute Pending Assignments
     T = 14;
                                       ! Clock Cycle 14
                                      ! Execute Assignment
     next;
     PHI1 = hi:
                                       ! Phase 1
     PHI2 = 10;
                                      ! Of Clock Cycle 14
                                      ! Execute Fending Assignments
     next;
     PHI1 = lo;
                                      ! Phase 2
                                      ! Of Clock Cycle 9
     PHI2 = hi:
     ASN = hi;
                                      ! Deactivate Address Strobe
                                      ! Deactivate Lower Data Strobe
     LISN = hi;
     UDSN = hi:
                                      ! Deactivate Upper Data Strobe
                                      ! Increment Program Counter
     PC = PC + 2;
     IR = PFR;
                                       ! Place Contents Of Prefetch
                                       ! Register Into Instruction
                                       ! Register
                                       ! Deactivate Data Transfer
     DITACKN = hi;
                                      ! Acknowledge(Added)
                                      ! Execute Pending Assignments
     next:
     T = 0
     )
                                      ! JMP (A0)
=: مسزر
     ! Phase 1 Of
     PHI1 = hi;
     FHI2 = 10;
                                       ! Clock Cycle 0
                                      ! Place Data Bus In A High Impedance
     DBUS = 0xffff;
                                       ! Memory Read
     RW = hi:
                                     ! Nisable Address Bus Buffer
     ADENABLE = 10:
                                       ! Disable Data Bus Ruffer
     DIBENABLE = 10:
     IABUS = PC;
                                      ! Place PC On Internal Address
                                       ! Bus
     next;
                                       ! Execute Pending Assignments
                                       ! Phase 2 Of
     PHI1 = lo;
     PHI2 = hi:
                                      ! Clock Cycle O
     ABENABLE = hi;
                                      ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                      ! Gate Internal Address Bus
                                       ! Into External Address Ruffer
     FCMODE = SRMODE;
                                      ! User Mode
     FCSPACE = 2;
                                       ! Accessing Program
```

(

```
! Execute Pending Assignments
next;
ABUS = EXABUF;
                                ! Address Flaced On Bus(Added)
next;
                                ! Execute Pending Assignments
T = 1:
                                ! Clock Cycle 1
next;
                                ! Execute Assignment
PHI1 = hi:
                                ! Phase 1 Uf
                                ! Clock Cycle 1
PH12 = 10;
                                ! Assert Address Strobe
ASN = 10;
LISN = lo;
                                ! Assert Lower Data Strobe
                                ! Assert Upper Data Strobe
UIISN = lo;
IABUS = AEOJ;
                                ! hove Jump Address From ACOJ
                                ! To Internal Address Buffer
                                ! Enable Data Bus
DBENABLE = hi:
                                ! Execute Pending Assignments
next;
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 1
PHI2 = hi;
PC = IABUS;
                               ! Place Jump Address Into Program
                                ! Counter
next;
1 = 2;
                                ! Clock Cycle 2
                                ! Execute Assignment
next;
                                ! Phase 1
PHI1 = hi;
                               ! Of Clock Cycle 2
PHI2 = 10;
while DTACKN eql hi
                                ! Wait For Memory To Place
                                ! Data On The Rus
                                ! Execute Impending Assignments
     next;
     PHI1 = lo;
                                ! Phase 2
                                ! Of Clock Cycle 2
     PHI2 = hi;
     next:
                                ! Execute Assignments
     T = 3;
                                ! Clock Cycle 3
                                ! Execute Assignment
     next;
                                ! Phase 1
     PHI1 = hi;
                                ! Of Clock Cycle 3
     PHI2 = 10;
     Libus<15:8> = MEABUS];
                                ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                                ! On Data Bus And
                                ! Asserts DTACKN(Added)
     DITACKN = 10;
                                ! Execute Pending Assignments
     next;
     T = 2
                               ! Return To Phase 2
                                ! Of Clock Cycle 2
     );
```

```
next;
                                 ! Execute Impending Assignments
T = 3:
                                 ! Clock Cycle 3
next;
                                 ! Execute Assignment
PHI1 = lo;
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 3
EXDRUF = DRUS;
                                 ! Instruction On Rata Bus
                                 ! Is Placed In External Data
                                 ! Bus Buffer
next;
                                 ! Execute Pending Assignments
! Clock Cycle 4
T = 4:
                                 ! Execute Assignment
next;
PHI1 = hi;
                                 ! Phase 1
                                 ! Of Clock Cycle 4
PHI2 = lo;
next;
PFR = EXDBUF;
                                 ! The Contents Of The External
                                 ! Data Bus Buffer Are Placed
                                 ! In Prefetch Register
next;
                                 ! Execute Pending Assignments
FHI1 = 10;
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 4
ASN = hi;
                                 ! Deactivate Address Strobe
LDSN = ni;
                                 ! Deactivate Lower Data Strobe
UISN = hi;
                                 ! Deactivate Upper Data Strobe
DTACKN = hi;
                                 ! Deactivate Data Transfer
                                 ! Acknowledge(Added)
next:
T = 5;
                                 ! Clock Cycle 5
next;
                                 ! Execute Previous Assignment
                                 ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                                 ! Clock Cycle 5
                                 ! Memory Read
RW = hi;
                                 ! Disable Address Bus Buffer
ADENABLE = 10;
                                 ! Disable Data Bus Buffer
DBENABLE = 10;
                                 ! Place PC On Internal Address
1ABUS = FC;
                                 ! Bus
                                 ! Execute Pending Assignments
next;
PHI1 = lo;
                                 ! Phase 2 Of
                                 ! Clock Cycle 5
PHI2 = hi;
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2:
                                 ! Accessing Program
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
                                 ! Into External Address Buffer
next;
```

```
ABUS = EXABUF;
                             ! Address Placed On Bus(Added)
                             ! Execute Pending Assignments
next;
T = 6;
                             ! Clock Cycle 6
                             ! Execute Assignment
next;
                             ! Phase 1 Of
PHI1 = hi;
PHI2 = 10:
                            ! Clock Cycle 6
                            ! Assert Address Strobe
ASN = lo;
                            ! Assert Lower Data Strobe
LDSN = lo;
UNSN = lo;
                            ! Assert Upper Data Strobe
DHENABLE = hi;
                             ! Enable Data Bus
                             ! Execute Pending Assignments
next;
PHI1 = lo;
                             ! Phase 2
                             ! Of Clock Cycle 6
PHI2 = hi;
next;
                             ! Execute Pending Assignments
T = 7;
                             ! Clock Cycle 7
                             ! Execute Assignment
next:
                             ! Phase 1
PHI1 = hi;
                             ! Of Clock Cycle 7
PHI2 = 10;
                             ! Wait For Memory To Place
while DTACKN eql hi
                             ! Data On The Bus
                             ! Execute Impending Assignments
    sext;
                             ! Phase 2
    FHI1 = lo;
                            ! Of Clock Cycle 7
    PHI2 = hi:
                            ! Execute Assignments
    next;
    ! Clock Cycle 8
    T = 8;
                             ! Execute Assignment
    next;
    PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 8
    PHI2 = 10;
    DBUS<15:8> = MEABUSD;
                             ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                             ! On Data Bus And
                             ! Asserts DTACKN(Added)
    DITACKN = 10;
                             ! Execute Pending Assignments
    next:
    ! Return To Phase 2
    T = 7
                            ! Of Clock Cycle 7
    );
                            ! Execute Impending Assignments
    next;
! Clock Cycle 8
T = 8;
                             ! Execute Assignment
next;
```

```
! Phase 2
     PHI1 = lo;
                                          ! Of Clock Cycle 8
     PHI2 = hi;
                                          ! Instruction On Data Bus
     EXBRUF = BBUS;
                                          ! Is Placed In External Data
                                          ! Bus Ruffer
                                          ! Execute Fending Assignments
     next;
     ! Clock Cycle 9
     T = 9;
                                          ! Execute Assignment
     next;
                                          ! Phase 1
     PHI1 = hi;
                                          ! Of Clock Cycle 9
     PHI2 = 10;
                                          ! The Contents Of The External
     PFR = EXDBUF:
                                          ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
                                          ! Execute Pending Assignments
     next;
                                          ! Phase 2
     PHI1 = lo;
                                          ! Of Clock Cycle 9
     PHI2 = hi;
                                          ! Deactivate Address Strobe
     ASN = hi;
                                          ! Deactivate Lower Data Strobe
     LIISN = hi;
                                          ! Deactivate Upper Data Strobe
     UDSN = hi;
                                          ! Increment Program Counter
     PC = PC + 2;
                                          ! Place Contents Of Prefetch
     IR = PFR;
                                          ! Register Into Instruction
                                          ! Register
                                          ! Deactivate Data Transfer
     DTACKN = hi;
                                          ! Acknowledge(Added)
                                          ! Execute Pending Assignments
     next;
                                          ! Reset Clock Cycle Counter
     T = 0
decode_execute_prefetch :=
                       case If
                           _Ox31c1: move
                                          ! MDVE.W D1,$2004 [2008]
                                          ! AND.W #$DFFF.SR
                           0x027c: andi
                                          ! JMP (AO) If IR = Octal Value
                           047320: jmp
                       esac
                       )
main :=
    power_on_initialize;
    fetch_initial_instruction;
    while READY eql hi
          decode_execute_prefetch
```

```
/*
                                                */
11
    MOTOROLA MC68000 MODEL OF THE HOVE.W A1, D3 INSTRUCTION
                                                */
/*
                                                11
/*
                                                */
                                                */
/*
              Structure Declarations
                                                */
/*
state
/*
                                                */
/*
           M68000 Programming Registers
                                                */
/*
                                                */
DE0:73<31:0>,
                  ! 8 Data Registers
AE0:63<31:0>,
                   ! 7 Address Registers
                  ! User Stack Pointer
UA7<31:0>,
SA7<31:0>,
                   ! System Stack Pointer
PC<31:0>,
                  ! Program Counter
SR<15:0>.
                   ! Status Register
/*
                                                */
/*
                                                */
           Temporary Internal Registers
/*
                                                x/
PFR<15:0>,
                   ! Frefetch Register
IR<15:0>,
                   ! Instruction Register
FC<210>,
                   ! Function Code Register
EXDBUF(15:0),
                  ! External Data Bus Buffer Register
EXABUF<23:1>,
                   ! External Address Bus Buffer Register(changed)
ALUBUF1 (31:0),
                   ! ALU Buffer 1
                   ! ALU Buffer 2
ALUBUF2<31:0>,
DTEMP<15:0>,
                   ! Temporary Data Storage
DISREG<31:0>,
                   ! Temporary Displacement Storage
SRTEMP<15:0>,
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPADR<31:0>,
                    Temporary Cycle Address Storage
                    (Exception Processing)
ACTYPE<15:0>.
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>,
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:0>.
                         ! Temporary Address Storage For
                         ! Exception Handler Routine
T<7:0>,
                         ! Clock Cycle Counter
RESET.
                      ! Reset Flip-Flop
HALT,
                      ! Halt Flip-Flop
RW,
                      ! Read/Write Flip-Flop
ADENABLE.
                      ! Address Bus Buffer Enable
DIBENABLE,
                      ! Data Bus Buffer Enable
ASN,
                      ! Address Strobe Flip-Flop
                      ! Lower Data Strobe Flip-Flop
LUSN.
UIISN,
                      ! Upper Data Strobe Flip-Flop
LITACKN.
                      ! Nata Transfer Acknowledge Flip-Flop
COUT.
                      ! Carry Flip-Flop
EXCEPT,
                      ! Exception Processing Flip-Flop
REALLY.
                      ! Ready Flip-Flop
/*
/x
                                                               */
       Model transformation modifications:
/*
                                                              */
/*
           1) CDL decoder structure nonexistent in ISF' and un-
/*
       necessary for model. Eliminated.
                                                              11/
/*

 Multi-phase clock structure nonexistent in ISP'.

                                                              */
/*
       Operations on registers will provide its equivalent.
                                                              */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
       register will provide its equivalent.
/*
                                                              */
/*
           4) The declared bus structures are modeled with registers */
       without loss of model accurracy. This done to maintain model
/×
                                                              */
/*
       equivalency and simplicity.
                                                              */
/*
           5) The memory word length was reduced from 16 to 8 bit
       words to coincide with the ECR's 32-Kbyte memory, to agree with*/
/*
1*
       their PC incrementation, and to enable the use of existing
                                                              */
/*
                                                              */
       MC68000 assembler and linker/loader models. The memory was
/*
       also reduced from 8 Mwords to 32 Kbytes.
                                                              1/
/*
                                                              */
1ABUS<31:0>,
                         ! Internal Address Bus
IDBUS<31:0>,
                         ! Internal Data Bus
                      ! Power Switch
SWITCH,
                      ! Phase 1 Of Two-Phase Clock
PHI1,
                      ! Phase 2 Of Two-Phase Clock
FH12;
port
/ x
                                                              X/
/*
              External Address and Data Bus
                                                              */
· /#
                                                               1/
! External Data Bus
DBUS(15:0>,
                        ! External Address Bus(changed)
ABUS<23:1>;
```

```
/*
                                                         */
/x
                                                         */
                Register Subfields
/*
                                                         1/
PCADDR
         = PC<23:0>,
                      ! Program Counter Address Field
SRTRACE
         = SR<15>,
                      ! Trace Bit
SRMODE
         = SR<13>,
                      ! Mode Selection Bit
SRCARRY
         = SR<0>,
                      ! Carry Fit
SKOVER
         = SR<1>,
                      ! Overflow Bit
SRZERO
         = SR<2>.
                      ! Zero Bit
                      ! Negative Bit
SRNEG
         = SR<3>,
SREX
         = SR<4>.
                      ! Extend Bit
SRMASK
         = SR<10:8>,
                      ! Interrupt Mask
FCSPACE
         = FC(1:0),
                      ! hemory access Address Space
FCMODE
         = FC(2),
                      ! User/Supervisor Mode Bit
                      ! PC Low Word
FCLOW
         = PC(15:0),
FCHI
                      ! PC High Word
         = PC(31:15),
A1LWORD
         = AE13<15:0>,
                      ! A[1] Low Word
DOLWORT:
         = D[0](15:0).
                      ! DEO3 Law Word
                      ! DE13 Low Word
DILWORD
         = DE13<15:0>,
         = DC23<15:0>,
D2LWORD
                      ! N[2] Low Word
T/3LWORD
         = D[33<15:0>,
                      ! BE33 Low Word
                      ! DE43 Low Word
DALWORD
         = DE4J<15:0>,
DISLUCKE
         = 10050<15:0>,
                      ! DE53 Low Word
DALWORD
         = DE63<15:0>.
                      ! DE63 Low Word
D7LWORD
         = DE73<15:0>,
                      ! DE70 Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
         = HANADR<15:0>, ! HANADR Low Word
HANADIRLOW
HANADEHI
         ≈ HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPAURHI
         = TEMPADR<31:16>;! TEMPADR High Word
Memory
/*
                                                         */
/*
                 16K 16-Bit Word Internal Memory
                                                         */
/*
                                                         */
MEO:327673<7:0>;
DIGCTO
/*
/*
                Logic Level Macros
                                                         */
```

format

٩

```
/*
= 0 1,
l٥
hi
    = 1 2,
    = 0 %,
off
on
    = 1 %,
clear = 0 %;
/*
                                                        */
                                                        */
/* Power On and Initialization. This process was not modeled but is
/*
   added to initialize signals and registers.
                                                        */
/*
                                                        */
power_on_initialize :=
                                ! Turn Power On
      SWITCH = on;
                                ! Execute Assignment
      next;
                                ! System Not Ready
      REALLY = lo:
                                ! Assert Reset For
      RESET = lo;
                                ! 100 Miliseconds(Active Low)
      delay(100);
      RESET = hi;
                                ! Deactivate Reset
                                ! Execute Fending Assignments
      next;
      ASN = hi;
                                ! Initialize Address Strobe
      LUSN = hi:
                                ! Initialize Lower Data Strobe
      UDSN = hi;
                                ! Initialize Upper Data Strobe
      DITACKN = h1;
                                ! Initialize Bata Transfer Acknowledge
                                ! Initialize Read/Write(Read On High)
      RW = hi:
                               ! Flace Nata Bus In High Impedance State
      DBUS = 0xffff;
      M[0x100a] = 0xff;
                                 ! Place Memory Locations Following The
      M[0:100b] = 0:ff:
                                  ! JMF Instruction In A High State
      HALT = hi:
                                ! Initialize Halt Flip-Flop(Active
                                ! Low)
      T = 0;
                                ! Initialize Clock Cycle Counter
      READY = hi;
                                ! System Ready
      /*
                                                        */
      /*
           Routine Initialization Fer Hamby and Guillory
                                                        */
                                                        */
      /*
      ! Place Hex 5555555 Into AC1]
      A[1] = 0 \times 555555555;
      A[0] = 0 \times 1004;
                                ! Place Hex 1004 Into A[0]
      FC = 0x1000;
                                ! Place Hex 1000 Into Program Counter
      next
                                ! Execute Assignments
*/
/*
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                        */
/* to retrieve modeled instructions for simulation and analysis. It
  was fashsioned from the Read Cycle described by Hamby and Guillory */
```

```
/* on page VI-15 of their thesis.
                                                          */
/*
fetch_initial_instruction :=
     ! Phase 1 Of
     FHI1 = h1;
    PHI2 = 10;
                                    ! Clock Cycle 0
     RW = hi;
                                    ! Memory Read
                                    ! Disable Address Bus Buffer
     ADENABLE = lo;
     DBENABLE = 10;
                                    ! Disable Data Bus Buffer
     TABUS = PC:
                                    ! Place FC On Internal Address
                                    ! Bus
                                    ! Execute Pending Assignments
    next;
     PHI1 = lo;
                                    ! Phase 2 Of
                                    ! Clock Cycle 0
    PHI2 = hi;
     ADENABLE = hi;
                                    ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                    ! Gate Internal Address Bus
                                    ! Into External Address Buffer
     FCMODE = SRMODE;
                                    ! User Mode
    FCSPACE = 2;
                                    ! Accessing Program
                                    ! Execute Impending Assignments
     next;
     ARUS = EXABUF;
                                    ! Address Placed On Rus(Added)
                                    ! Execute Pending Assignments
     next;
     ! Clock Cycle 1
     T = 1;
     next;
                                    ! Execute Assignment
     PHI1 = hi;
                                    ! Phase 1 Of
                                    ! Clock Cycle 1
     PHI2 = 10;
                                    ! Assert Address Strobe
     ASN = lo;
                                    ! Assert Lower Data Strobe
     LISN = lo;
                                    ! Assert Upper Data Strobe
     UPSN = 10;
     DBENABLE = hi;
                                    ! Enable Data Bus
                                    ! Execute Pending Assignments
     next;
                                    ! Phase 2
     PHI1 = 10;
                                    ! Of Clock Cycle 1
     PHI2 = hi;
                                    ! Execute Pending Assignments
     T = 2:
                                    ! Clock Cycle 2
                                    ! Execute Assignment
     next;
                                    ! Phase 1
    FHII = hi;
                                    ! Of Clock Cycle 2
    PHI2 = 10:
     while ITACKN eql hi
                                    ! Wait For Memory To Place
                                    ! Data On The Rus
```

```
next:
                               ! Execute Impending Assignments
     PHI1 = lo;
                                ! Phase 2
     PHI2 = hi;
                               ! Of Clock Cycle 2
     next;
                               ! Execute Assignments
     ! Clock Cycle 3
     next;
                                ! Execute Assignment
     PHI1 = hi;
                               ! Phase 1
     FRI2 = 10;
                                ! Of Clock Cycle 3
     DBUS<15:8> = MEABUSD;
                              ! Memory Flaces Instruction
     IDBUS <7:6> = MEABUS + 13;
                               ! On Data Rus And
     DIACKN = 10;
                               ! Asserts DTACKN(Added)
    next;
                               ! Execute Pending Assignments
     T = 2
                                ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
     next;
                               ! Execute Impending Assignments
T = 3;
                               ! Clock Cycle 3
next;
                                ! Execute Assignment
PHI1 = 10;
                               ! Phase 2
PH12 = hi
                               ! Of Clock Cycle 3
EXDBUF = DBUS;
                               ! Instruction On Data Rus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
f'HI2 = lo;
                               ! Of Clock Cycle 4
PFR = EXDBUF:
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Fending Assignments
FHI1 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 4
ASN = hi;
                               ! Deactivate Address Strobe
LIISN = hi;
                               ! Neactivate Lower Nata Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
IR = PFR;
                                ! Contents Of Prefetch Register
                                ! Are Placed Into Instruction
                                ! Register
```

```
DTACKN = hi;
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
                                         ! Increment Program Counter
     PC = PC + 4;
     next;
                                         ! Execute Pending Assignments
     T = 0
                                         ! Reset Clock Cycle Counter
                                         ! MOVE.W A1, D3
move :=
     PHI1 = hi:
                                         ! Phase 1 Of
     PHI2 = 10;
                                         ! Clock Cycle 0
     DBUS = OXFFFF;
                                         ! Place bata Bus In High Impedance
     RW = hi;
                                         ! Memory Read
     ADENABLE = lo;
                                         ! Disable Address Bus Buffer
     DRENABLE = 10;
                                         ! Disable Data Bus Buffer
     IABUS = FC;
                                         ! Place PC On Internal Address
                                         ! Bus
     IDBUS = AILWORD;
                                         ! Place Low Word From A[1] Onto
                                         ! Internal Data Bus
                                         ! Execute Pending Assignments
     next;
     PHIi = lo;
                                         ! Phase 2 Of
                                         ! Clock Cycle O
     PHI2 = hi:
     ADENABLE = hi;
                                         ! Enable Address Bus Buffer
     EXARUF = IARUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
     FCMODE = SRMODE;
                                         ! User Mode
     FCSPACE = 2;
                                         ! Accessing Program
                                         ! Clear Status Register Carry Bit
     SRCARRY = 10;
                                         ! Clear Status Register Overflow Bit
     SROVER = 10;
     SRZERO = lo;
                                         ! Clear Status Register Zero Bit
                                         ! Clear Status Register Negative Bit
     SRNEG = 10;
                                         ! Place PC On Address Rus (Added)
     ABUS = IABUS;
     D3LWORD = IDBUS;
                                         ! Place Nata From Internal Data Rus
                                         ! Into Low Word Of D[3]
                                         ! Execute Impending Assignments
     next;
     T = 1;
                                         ! Clock Cycle 1
     next;
                                         ! Execute Assignment
                                         ! Phase 1 Of
     PHI1 = hi;
     f'HI2 = 10;
                                         ! Clock Cycle 1
     ASN = lo;
                                         ! Assert Address Strobe
     LDSN = 10;
                                         ! Assert Lower Nata Strobe
     UDSN = lo;
                                         ! Assert Upper Data Strobe
                                         ! Enable Data Bus
     DBENABLE = hi;
     if D3LWORD eal 0
                                         ! Set Status Register Zero Bit
```

```
SRZERO = hi;
                             ! If Moved Data Is Zero
                              ! Execute Pending Assignments
next:
PHI1 = 10;
                              ! Phase 2
PHI2 = hi;
                              ! Of Clock Cycle 1
                              ! Set Status Register Negative
if DE33<15>
  SRNEG = hi;
                              ! Rit If Moved Data Is Negative
next;
                              ! Execute Pending Assignments
T = 2;
                              ! Clock Cycle 2
next;
                              ! Execute Assignment
                              ! Phase 1
PHI1 = hi;
FHI2 = 10;
                             ! Of Clock Cycle 2
while BTACKN eql hi
                             ! Wait For Memory To Place
                              ! Data On The Bus
                              ! Execute Impending Assignments
    next;
    FHI1 = 10;
                             ! Phase 2
    PHI2 = hi:
                              ! Of Clock Cycle 2
                              ! Execute Assignments
    next:
    ! Clock Cycle 3
    next;
                              ! Execute Assignment
    PHI1 = hi;
                              ! Phase 1
                              ! Of Clock Cycle 3
    PHI2 = 10;
    DBUS<15:8> = MEABUS3;
                              ! Memory Flaces Instruction
    DBUS<7:0> = MEABUS + 10;
                              ! On Data Bus And
                              ! Asserts ITACKN(Added)
    INTACKN = 10;
                              ! Execute Pending Assignments
    next;
    T = 2
                              ! Return To Phase 2
                              ! Of Clock Cycle 2
    );
                             ! Execute Impending Assignments
    next;
T = 3:
                              ! Clock Cycle 3
next;
                              ! Execute Assignment
                             ! Phase 2
PHI1 = lo;
                              ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DBUS;
                              ! Instruction On Data Bus
                              ! Is Placed In External Data
                              ! Bus Euffer
                              ! Execute Pending Assignments
next;
T = 4;
                              ! Clock Cycle 4
```

```
! Execute Assignment
     next;
                                         ! Phase 1
     PHI1 = hi;
                                         ! Of Clock Cycle 4
     FH12 = 10:
                                         ! The Contents Of The External
     PFR = EXDBUF;
                                         ! Data Bus Buffer Are Placed
                                         ! In Prefetch Register
                                         ! Execute Fending Assignments
     next;
     PHI1 = 10;
                                         ! Phase 2
                                         ! Of Clock Cycle 4
     PHI2 = hi;
     ASN = hi;
                                         ! Deactivate Address Strobe
                                         ! Deactivate Lower Data Strobe
     LISN = hi;
                                         ! Deactivate Upper Data Strobe
     UDSN = hi;
     IR = PFR;
                                         ! Contents Of Prefetch Register
                                         ! Are Placed Into Instruction
                                         ! Register
     DTACKN = hi;
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
     PC = PC + 2;
                                         ! Increment Program Counter
     next;
                                         ! Execute Impending Assignments
     T = 0
                                         ! Reset Clock Cycle Counter
     )
=: aut.
                                         ! JMP (A0)
     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                         ! Clock Cycle 0
                                         ! Place Data Bus In A High Impedance
     IBUS = 0xffff;
                                         ! Memory Read
     RW = hi;
     ADENABLE = 10;
                                         ! Disable Address Bus Buffer
                                         ! Disable Data Bus Buffer
     DIBENABLE = 10;
                                         ! Place PC On Internal Address
     IABUS = PC;
     next;
                                         ! Execute Pending Assignments
     FHI1 = lo;
                                         ! Phase 2 Of
                                         ! Clock Cycle 0
     PHI2 = hi;
     ADENABLE = hi:
                                         ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
     FCMODE = SRMODE;
                                         ! User Mode
     FCSFACE = 2;
                                         ! Accessing Program
     next;
                                         ! Execute Pending Assignments
     ABUS = EXABUF;
                                         ! Address Flaced On Bus(Added)
                                         ! Execute Pending Assignments
     next;
     ! Clock Cycle 1
     T = 1;
                                         ! Execute Assignment
     next;
```

THE CONTROL OF THE PROPERTY OF THE PARTY OF

```
PHI1 = hi;
                               ! Phase 1 Of
PH12 = 10:
                               ! Clock Cycle 1
ASN = 10;
                               ! Assert Address Strobe
LISN = lo:
                               ! Assert Lower Data Strobe
UDSN = lo:
                               ! Assert Upper Data Strobe
TARUS = A[0];
                               ! hove Jump Address From A[0]
                               ! To Internal Address Buffer
                               ! Enable Data Bus
DRENABLE = hi;
next;
                              ! Execute Pending Assignments
PHI1 = 10;
                              ! Phase 2
                               ! Of Clock Cycle 1
PHI2 = hi;
PC = IABUS;
                               ! Place Jump Address Into Program
                               ! Counter
next;
T = 2;
                               ! Clock Cycle 2
next;
                               ! Execute Assignment
· PHII = hi:
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 2
                               ! Wait For Memory To Place
while DTACKN eql hi
                              ! Data On The Rus
     (
                              ! Execute Impending Assignments
     next;
    · PHI1 = 16;
                              ! Phase 2
                              ! Of Clock Cycle 2
     PHI2 = hi;
                              ! Execute Assignments
     next;
     ! Clock Cycle 3
     next;
                              ! Execute Assignment
                              ! Phase 1
     FHI1 = hi;
                              ! Of Clock Cycle 3
     PH12 = 10;
     DRUS<15:8> = MCABUSJ;
                              ! Memory Flaces Instruction
     DBUS<7:0> = MEABUS + 13;
                              ! On Data Bus And
     DITACKN = 10;
                               ! Asserts DTACKN(Added)
     next;
                               ! Execute Pending Assignments
     ! Return To Phase 2
     T = 2
                               ! Of Clock Cycle 2
     );
     next;
                              ! Execute Impending Assignments
T = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
                               ! Phase 2
FHI1 = 10;
```

```
PHI2 = hi;
                                 ! Of Clock Cycle 3
EXDRUF = DRUS;
                                  ! Instruction On Nata Bus
                                 ! Is Placed In External Data
                                 ! Bus Buffer
next;
                                  ! Execute Pending Assignments
T = 4;
                                 ! Clock Cycle 4
                                 ! Execute Assignment
next;
PHI1 = hi;
                                 ! Phase 1
PHI2 = 16
                                 ! Of Clock Cycle 4
next;
                                 ! The Contents Of The External
PFR = EXDBUF;
                                 ! Data Bus Buffer Are Placed
                                  ! In Prefetch Register
next;
                                  ! Execute Pending Assignments
PH11 = 10;
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 4
                                 ! Deactivate Address Strobe
ASN = hi;
                                 ! Deactivate Lower Data Strobe
LDSN = hi;
                                 ! Deactivate Upper Data Strobe
UDSN = hi:
                                 ! Deactivate Data Transfer
DTACKN = hi;
                                 ! Acknowledge(Added)
T = 5;
                                 ! Clock Cycle 5
next;
                                  ! Execute Previous Assignment
PHI1 = hi;
                                 ! Phase 1 Of
                                  ! Clock Cycle 5
PHI2 = 10;
RW = hi:
                                 ! Memory Read
ADENABLE = lo;
                                 ! Disable Address Bus Buffer
DEENABLE = 10;
                                  ! Disable Data Bus Buffer
IABUS = PC;
                                  ! Flace PC On Internal Address
                                  ! Rus
                                  ! Execute Pending Assignments
next;
                                 ! Phase 2 Of
PHI1 = 10;
                                 ! Clock Cycle 5
PHI2 = hi;
ADENABLE = hi:
                                 I Enable Address Bus Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2;
                                 ! Accessing Program
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
                                 ! Into External Address Buffer
next;
ABUS = EXABUF;
                                 ! Address Placed On Rus(Added)
                                 ! Execute Pending Assignments
next;
! Clock Cycle 6
T = 6:
                                  ! Execute Assignment
next;
```

```
! Phase 1 Of
PHI1 = hi;
                               ! Clock Cycle 6
PHI2 = 10;
                               ! Assert Address Strobe
ASN = 10;
LISN = lo;
                               ! Assert Lower Data Strobe
UDSN = lo;
                               ! Assert Upper Data Strobe
                               ! Enable Data Bus
DBENABLE = hi;
                               ! Execute Pending Assignments
next:
FHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 6
PHI2 = hi;
next;
                               ! Execute Pending Assignments
T = 7;
                               ! Clock Cycle 7
                               ! Execute Assignment
next;
PHI1 = hi:
                               ! Phase 1
                               ! Of Clock Cycle 7
PHI2 = 10;
                               ! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
    next;
                               ! Execute Impending Assignments
    PHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 7
    PHI2 = hi:
    next;
                               ! Execute Assignments
    T = 8:
                               ! Clock Cycle 8
                               ! Execute Assignment
    next;
    PHI1 = hi;
                               ! Phase 1
    PHI2 = 10;
                               ! Of Clock Cycle 8
    DBUS<15:8> = MEABUSD;
                               ! Memory Places Instruction
    I(BUS<7:0> = MEABUS + 13;
                               ! On Data Rus And
    DTACKN = lo;
                               ! Asserts DTACKN(Added)
    next;
                               ! Execute Pending Assignments
    T = 7
                              ! Return To Phase 2
                              ! Of Clock Cycle 7
     );
     next;
                              ! Execute Impending Assignments
T = 8:
                               ! Clock Cycle 8
                               ! Execute Assignment
next;
                               ! Phase 2
PHI1 = lo;
                               ! Of Clock Cycle 8
PHI2 = hi;
EXDRUF = DRUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
```

```
T = 9;
                                          ! Clock Cycle 9
     next;
                                          ! Execute Assignment
     PHI1 = hi;
                                          ! Phase 1
     PHI2 = 10:
                                          ! Of Clock Cycle 9
     PFR = EXDBUF;
                                          ! The Contents Of The External
                                        · ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
     next;
                                          ! Execute Pending Assignments
      PHI1 = lo;
                                          ! Phase 2
     PHI2 = hi
                                          ! Of Clock Cycle 9
     ASN = hi;
                                          ! Deactivate Address Strobe
     LDSN = hi:
                                          ! Deactivate Lower Data Strobe
     UDSN = hi;
                                          ! Deactivate Upper Data Strobe
     PC = PC + 2;
                                          ! Increment Program Counter
     IR = PFR;
                                          ! Place Contents Of Prefetch
                                          ! Register Into Instruction
                                          ! Register
                                          ! Deactivate Data Transfer
     DTACKN = hi;
                                          ! Acknowledge(Added)
     next:
                                          ! Execute Pending Assignments
     T = 0
                                          ! Reset Clock Cycle Counter
andi :=
                                          ! AND.W ##DFFF,SE
    SRMODE = 16:
                                          ! Set Status Register To
    IR<15:8> = MCPC3:
                                          ! User Mode And Prefetch
    IR<7:0> = MEFC + 13;
                                          ! Next Instruction
    next;
    PC = PC + 2;
                                          ! Increment PC
    T = 5;
                                          ! Requires 6 Clock Cycles
    next;
    T = 0
    )
decode_execute_prefetch :=
                       case IR
                            0x3609: move
                                          ! MOVE.W A1.D3
                           0x027c: andi
                                          ! AND.W #DFFF,SR
                            047320: JBD
                                          ! JMF (AO) If IR = Octal Value
                       esac
main :=
    power_on_initialize;
    fetch_initial_instruction;
    while REABY eql hi
```

(
decode_execute_prefetch
)

```
*/
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W (A1), D2 INSTRUCTION
                                               */
                                               */
/*
/*
                                               */
/*
              Structure Declarations
                                               */
/x
                                               */
state
/*
                                               */
/*
                                               */
           M68000 Programming Registers
/*
                                               */
DE0:73<31:0>,
                  ! 8 Nata Registers
A[0:63<31:0>,
                  ! 7 Address Registers
UA7<31:0>,
                  ! User Stack Pointer
SA7<31:0>.
                  ! System Stack Pointer
PC<31:0>,
                  ! Program Counter
SR<15:0>,
                  ! Status Register
/*
                                               */
/*
           Temporary Internal Registers
                                               */
/*
                                               */
PFR<15:0>,
                  ! Frefetch Register
                  ! Instruction Register
IR<15:0>.
FC<2:0>,
                  ! Function Code Register
                  ! External Data Bus Buffer Register
EXDBUF<15:0>,
EXABUF<23:1>.
                  ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,
                  ! ALU Buffer 1
                  ! ALU Buffer 2
ALUBUF2<31:0>,
DTEMP<15:0>.
                  ! Temporary Data Storage
DISREG<31:0>,
                  ! Temporary Displacement Storage
SRTEMP<15:0>,
                  ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                  ! Temporary Instruction Register Storage
                  ! (Exception Processing)
                  ! Temporary Cycle Address Storage
TEMPADR<31:0>,
                   ! (Exception Processing)
ACTYPE<15:0>,
                  ! Temporary Access Type Storage
                  ! (Exception Processing)
VECADR<23:0>,
                  ! Temporary Vector Address Storage
                  ! (Exception Processing)
```

```
HANADR<31:0>,
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
                        ! Clock Cycle Counter
T<7:0>,
                     ! Reset Flip-Flop
RESET,
HALT,
                      ! Halt Flip-Flop
FiW.
                      ! Read/Write Flip-Flop
ADENABLE,
                      ! Address Rus Buffer Enable
DISENABLE,
                      ! Data Bus Buffer Enable
                      ! Address Strobe Flip-Flop
ASN,
LIISN,
                      ! Lower Data Strobe Flip-Flop
UDSN,
                      ! Upper Data Strobe Flip-Flop
DITACKN,
                      ! Data Transfer Acknowledge Flip-Flop
COUT.
                      ! Carry Flip-Floo
EXCEPT,
                      ! Exception frocessing Flip-Flop
READY.
                      ! Ready Flip-Flop
/*
                                                              */
/*
       Model transformation modifications:
                                                              */
/×
                                                              x/
/*
                                                              */
           1) CDL decoder structure nonexistent in ISP' and un-
       necessary for model. Eliminated.
/*
                                                              */
/*
           2) Multi-phase clock structure nonexistent in ISP'.
                                                              */
/#
       Operations on registers will provide its equivalent.
                                                              ¥/
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
/#
       register will provide its equivalent.
                                                              */
           4) The declared bus structures are modeled with registers */
/x
/*
       without loss of model occurracy. This done to maintain model
                                                              */
                                                              */
/x
       equivalency and simplicity.
           5) The memory word length was reduced from 16 to 8 bit
                                                              1/
11
/x
       words to coincide with the ECB's 32-Kbyte memory, to agree with*/
                                                              */
/*
       their PC incrementation, and to enable the use of existing
       MC68000 assembler and linker/loader models. The memory was
/*
                                                              */
/*
       also reduced from 8 Mwords to 32 Kbytes.
                                                              */
/x
                                                              */
! Internal Address Bus
IABUS(31:0),
IDBUS<31:0>,
                        ! Internal Data Rus
twait<7:0>,
                        ! Wait Cycle Counter
SWITCH,
                      ! Fower Switch
                      ! Phase 1 Of Two-Phase Clock
PHI1,
                      ! Phase 2 Of Two-Phase Clock
PHI2;
port
/*
                                                              */
/*
              External Address and Data Bus
                                                              */
                                                              1/
/*
DBUS<15:0>,
                        ! External Data Bus
```

```
ABUS<23:1>;
                        ! External Address Bus(changed)
format
人术在状态术效果或术术大术型扩大型状体状术大术广中或非术效素性效果或不能表现成成的或类型的成果或激素或激素的表现的现在形式的影响的影响。
/X
                                                            1/
/*
                                                             */
                 Register Subfields
/1
                                                             */
FCAUUR
          = FC(23:0),
                        ! Program Counter Address Field
SRTRACE
                        ! Trace Bit
          = SR<15>,
                        ! Mode Selection Bit
SAMODE
          = SR<13>,
                        ! Carry Bit
SRCARRY
          = SR<0>,
          = SR<1>,
SROVER
                        ! Overflow Bit
SRZERO
                        ! Zero Rit
          = SR<2>,
SKNEG
          = SR<3>.
                        ! Negative Bit
SREX
          = SR<4>,
                        ! Extend Bit
SRMASK
          = SR<10:8>,
                        ! Interrupt Mask
FCSF'ACE
          = FC<1:0>,
                        ! Memory Access Address Space
FCMODE
          = FC<2>,
                        ! User/Supervisor Mode Bit
F'CLOW
          = PC<15:0>.
                        ! FC Low Word
PCHI
          = PC<31:16>,
                        ! PC High Word
                        ! DEOJ Low Word
IJOLWORT
          = D[0]<15:0>.
DILWORD
                        ! DE13 Low Word
          = D[1]\langle 15:0\rangle.
1/2LWORD
          = D[2](15:0),
                        ! B[2] Low Word
                        ! DE3] Low Word
D3LWORD
          = D[3](15:0),
          = D[43<15:0>,
II4LWORD
                        ! DL43 Low Word
          = DE51<1510>,
                        ! DESI Low Word
DSLWORD
                        ! DE63 Low Word
IISLUORI
          = DE63<15:0>,
                        ! DE73 Low Word
DZLWORD
          = DE73<15:0>.
DISREGHWORD = DISREG(31:16),! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
          = HANADR<15:0>, ! HANADR Low Word
HANAIIRLOW
HANADRHI
          = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
          = TEMPADR<31:16>;! TEMPADR High Word
TEMPADRHI
memory
14
                                                             */
/*
                  16K 16-Rit Word Internal Memory
                                                             */
/*
                                                             */
ME0:327673<7:0>;
macro
/x
                                                             */
/*
                                                             */
                 Logic Level Macros
```

```
/*
= 0 %,
l٥
    = 1 %,
hi
off
    = 0 2.
ĊΠ
    = 1 %,
clear = 0 %;
*/
/*
                                                        */
/* Fower On and Initialization. This process was not modeled but is
                                                        x/
/* added to initialize signals and registers.
/*
                                                        */
power_on_initialize :=
      SWITCH = on;
                                 ! Turn Fower On
                                 ! Execute Assignment
      next;
      READY = 10;
                                 ! System Not Ready
      RESET = lo;
                                ! Assert Reset For
      delay(100);
                                ! 100 Miliseconds(Active Low)
      RESET = hi;
                                ! Deactivate Reset
                                ! Execute Pending Assignments
      next;
      ASN = hi:
                                ! Initialize Address Strobe
      LDSN = hi:
                                ! Initialize Lower Data Strobe
      UDSN = hi;
                                ! Initialize Upper Data Strobe
      DITACKN = hi;
                                ! Initialize Data Transfer Acknowledge
                                ! Initialize Read/Write(Read On High)
      RW = hi;
                                ! Place Nata Bus In High Impedance State
      DBUS = 0xffff;
                                ! Place memory Locations Following The
      ME0 \times 100 a = 0 \times ff;
      M[0\times100b] = 0\times ff;
                                  ! JMP Instruction In A High State
                                ! Initialize Halt Flip-Flop(Active
      HALT = hi;
                                ! Low)
      T = 0;
                                ! Initialize Clock Cycle Counter
      READY = hi;
                                 ! System Ready
      */
      /*
            Routine Initialization Per Hamby and Guillory
                                                        1/
      /*
                                                        */
      ME0x20003 = 0x55;
                                ! Initialize Memory Location
                                ! 2000 Hex To 5555 Hex
      ML0x20013 = 0x55;
      A[0] = 0 \times 1004;
                                ! Place Hex 1004 Into ACOJ
      A[1] = 0:2000;
                                ! Store Data At Hex 2000
      FC = 0x1000;
                                ! Place Hex 1000 Into Program Counter
                                ! Execute Assignments
      next
      )
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
```

```
/* to retrieve modeled instructions for simulation and analysis. It
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/* on page VI-15 of their thesis.
                                                           */
fetch_initial_instruction :=
     PHI1 = hi:
                                     ! Phase 1 Of
     FHI2 = 10;
                                     ! Clock Cycle 0
     RW = hi;
                                     ! Memory Read
     ADENABLE = 10;
                                     ! Disable Address Bus Buffer
     DBENABLE = 10;
                                     ! Disable Data Bus Buffer
     1ABUS = PC;
                                     ! Place PC On Internal Address
                                    ! Execute Pending Assignments
     next;
                                    ! Phase 2 Of
     PHI1 = lo;
                                    ! Clock Cycle 0
     PHI2 = hi;
     ADENABLE = hi;
                                    ! Enable Address Bus Buffer
     EXABUF = TABUS;
                                    ! Gate Internal Address Bus
                                    ! Into External Address Buffer
    FCMODE = SRMODE;
                                    ! User Mode
     FCSPACE = 2;
                                    ! Accessing Program
                                    ! Execute Impending Assignments
     next;
     ABUS = EXABUF:
                                    ! Address Flaced Un Bus(Added)
                                     ! Execute Pending Assignments
     next;
     T = 1:
                                     ! Clock Cycle 1
     next:
                                     ! Execute Assignment
     PHI1 = hi;
                                    ! Phase 1 Of
     PHI2 = 10;
                                    ! Clock Cycle 1
     ASN = lo;
                                    ! Assert Address Strobe
    LDSN = lo;
                                    ! Assert Lower Data Strobe
     UDSN = lo:
                                    ! Assert Upper Data Strobe
    DBENABLE = hi;
                                    ! Enable Data Rus
     next;
                                     ! Execute Pending Assignments
                                    ! Phase 2
     FHI1 = 10;
     PHI2 = hi;
                                     ! Of Clock Cycle 1
     next;
                                     ! Execute Pending Assignments
     T = 2;
                                     ! Clock Cycle 2
     next;
                                     ! Execute Assignment
     PHI1 = hi:
                                     ! Phase 1
     PHI2 = 10:
                                     ! Of Clock Cycle 2
```

着きらくとこうの意見されたのうなと言葉ないできてい

(.

```
! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
                               ! Execute Impending Assignments
    next;
                               ! Phase 2
     PHI1 = lo;
    PHI2 = hi;
                               ! Of Clock Cycle 2
                                ! Execute Assignments
     next;
     T = 3;
                                ! Clock Cycle 3
                               ! Execute Assignment
     next;
                                ! Phuse 1
     PHI1 = hi;
                               ! Of Clock Cycle 3
     PHI2 = 10;
     DBUS<15:8> = MEABUS3;
                               ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
                               ! Asserts DTACKN(Added)
     DTACKN = 10;
     next;
                                ! Execute Pending Assignments
     ! Return To Phase 2
     T = 2
                                ! Of Clock Cycle 2 ...
     );
     next;
                               ! Execute Impending Assignments
! Clock Cycle 3
T = 3:
next;
                                ! Execute Assignment
                                ! Phase 2
PHI1 = 10;
                                ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DRUS;
                                ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Bus Ruffer
                                ! Execute Pending Assignments
next;
! Clock Cycle 4
T = 4;
                                ! Execute Assignment
next;
PH11 = hi;
                                ! Phase 1
PHI2 = 10;
                                ! Of Clock Cycle 4
                                ! The Contents Of The External
PFK = EXDBUF;
                                ! Data Bus Buffer Are Placed
                                ! In Prefetch Register
                                ! Execute Pending Assignments
next;
                                ! Phase 2
PHI1 = lo;
                                ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                                ! Deactivate Address Strobe
LISN = Li:
                                ! Deactivate Lower Data Strobe
                                ! Deactivate Upper Data Strobe
UDSN = ha;
                                ! Contents Of Prefetch Register
IR = PFR;
```

```
! Are Placed Into Instruction
                                         ! Register
     DTACKN = hi:
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
     PC = PC + 4;
                                         ! Increment Program Counter
                                         ! Execute Pending Assignments
     next;
     T = 0
                                         ! Reset Clock Cycle Counter
                                         ! AND.W #$DFFF,SR
undi :=
                                         ! Effect Of Instruction
    SRMODE = 10;
                                         ! Prefetch Next Instruction
    IR<15:8> = MCPC3:
    IR<7:0> = MCPC + 13;
                                         ! Is To Set Status Register
    next:
                                           ! Increment Program Counter
       PC = PC + 2:
                                         ! Supervisor Bit To User
    T = 5;
    next;
                                         ! hode
    T = 0
                                         ! Requires 6 Clock Cycles
                                         ! MOVE.W (A1),D2
move :=
     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                         ! Clock Cycle 0
     DBUS = Oxffff;
                                         ! Place Data Bus In High Impedance
                                         ! Memory Read
     RW = hi;
                                         ! Disable Address Bus Kuffer
     ADENABLE = lo;
                                         ! Disable Data Bus Buffer
     DBENABLE = 10;
     IABUS = PC:
                                         ! Place PC On Internal Address
                                         ! Bus
                                         ! Execute Pending Assignments
     next;
                                         ! Phase 2 Of
     FHI1 = 10;
                                         ! Clock Cycle 0
     PHI2 = hi;
     ADENABLE = hi;
                                         ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
                                         ! User Mode
     FCMODE = SRMODE;
     FCSPACE = 2;
                                         ! Accessing Program
     ABUS = IABUS;
                                         ! Place PC On Address Bus
     next;
                                         ! Execute Impending Assignments
     ! Clock Cycle 1
     T = 1;
     next;
                                         ! Execute Assignment
                                         ! Phase 1 Of
     PHI1 = hi;
                                         ! Clock Cycle 1
     PHI2 = 10;
                                         ! Assert Address Strobe
     ASN = lo;
```

```
LDSN = lo;
                              ! Assert Lower Data Strobe
UI(SN = 10;
                              ! Assert Upper Data Strobe
DHENABLE = hi;
                              ! Enable Data Bus
next;
                              ! Execute Pending Assignments
PHI1 = lo:
                              ! Phase 2
FHI2 = t.i;
                              ! Of Clock Cycle 1
                              ! Execute Pending Assignments
next;
T = 2;
                              ! Clock Cycle 2
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1
                              ! Of Clock Cycle 2
PHI2 = 10;
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Data On The Bus
                             ! Execute Impending Assignments
    next;
    PHI1 = lo;
                             ! Phase 2
    FHI2 = hi;
                             ! Of Clock Cycle 2
    nexti
                             ! Execute Assignments
    T = 3;
                             ! Clock Cycle 3
                              ! Execute Assignment
    next;
                             ! Phase 1
    PHI1 = hi;
    fHI2 = 16;
                             ! Of Clock Cycle 3
    DBUS<15:8> = MEARUSD;
                             ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                             ! On Nata Bus And
    LITACKN = 10;
                              ! Asserts I:TACKN(Added)
    next;
                              ! Execute Fending Assignments
    T = 2
                              ! Return To Phase 2
                              ! Of Clock Cycle 2
    );
                              ! Execute Impending Assignments
    next;
T = 3:
                              ! Clock Cycle 3
next;
                              ! Execute Assignment
FHI1 = 10;
                             ! Phase 2
PHI2 = hi;
                              ! Of Clock Cycle 3
EXDBUF = DBUS;
                             ! Instruction On Data Bus
                             ! Is Placed In External Data
                              ! Bus Buffer
                            ! Execute Pending Assignments
next:
! Clock Cycle 4
T = 4;
```

これでは最近の意味をはいましたが、これには

```
next;
                                   ! Execute Assignment
PHI1 = hi;
                                   ! Phase 1
PHI2 = 10;
                                   ! Of Clock Cycle 4
PFR = EXBBUF;
                                   ! The Contents Of The External
                                   ! Nata Bus Buffer Are Flaced
                                   ! In Prefetch Register
                                   ! Execute Pending Assignments
next;
PHI1 = lo;
                                   ! Phase 2
PHI2 = hi;
                                   ! Of Clock Cycle 4
                                   ! Deactivate Address Strobe
ASN = hi;
LDSN = hi;
                                   ! Deactivate Lower Duta Strobe
                                   ! Deactivate Upper Data Strobe
UDSN = hi;
PC = PC + 2;
                                   ! Increment Program Counter
                                   ! Are Placed Into Instruction
                                    ! Register
DTACKN = hi:
                                    ! Deactivate Nata Transfer(Added)
                                    ! Acknowledge
next:
T = 5;
                                   ! Clock Cycle 5
next;
                                   ! Execute Previous Assignment
                                   ! Phase 1 Of
PHI1 = hi;
PH12 = 10;
                                   ! Clock Cycle 5
                                   ! Place Date Rus In High Impedance
IBUS = 0xffff;
RW = hi;
                                   ! Memory Read
                                   ! Disable Address Bus Buffer
ADENABLE = 10;
IIBENABLE = 10;
                                   ! Disable Data Bus Buffer
IABUS = AE1];
                                   ! Place A[1] On Internal Address
                                   ! Bus
next;
                                   ! Execute Pending Assignments
FHI1 = lo;
                                   ! Phase 2 Of
PHI2 = hi:
                                   ! Clock Cycle 5
                                   ! Enable Address Bus Buffer
ADENABLE = hi;
                                   ! Gate Internal Address Rus
EXABUF = IABUS;
                                   ! Into External Address Buffer
FCMODE = SRMODE;
                                   ! User Mode
FCSFACE = 1;
                                   ! Accessing Data
SRCARRY = 10;
                                   ! Clear Status Register Carry Bit
SROVER = 10;
                                   ! Clear Status Register Overflow Bit
                                   ! Clear Status Register Zero Bit
SRZERO = lo;
SRNEG = 10:
                                   ! Clear Status Register Negative Bit
ABUS = IABUS;
                                   ! Place PC On Address Bus (Added)
                                   ! Execute Impending Assignments
next:
T = 6;
                                   ! Clock Cycle 6
next;
                                   ! Execute Assignment
```

```
PHI1 = hi;
                                ! Phase 1 Of
                               ! Clock Cycle 6
PHI2 = 10;
                               ! Assert Address Strobe
ASN = 10;
LUSN = lo;
                               ! Assert Lower Data Strobe
UDSN = 10;
                               ! Assert Upper Data Strobe
DBENABLE = hi;
                               ! Enable Data Bus
                               ! Execute Pending Assignments
next;
                               ! Phase 2
FHI1 = 10;
                               ! Of Clock Cycle 6
PHI2 = hi;
                               ! Execute Pending Assignments
next;
T = 7;
                               ! Clock Cycle 7
                                ! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi;
                               ! Of Clock Cycle 7
FHI2 = 10;
                               ! Wait For Memory To Flace
while DTACKN eql hi
                               ! Nata On The Rus
                               ! Execute Impending Assignments
     next;
     PHI1 = lo;
                               ! Phase 2
     PHI2 = hi;
                               ! Of Clock Eycle 7
                               ! Execute Assignments
     next;
     T = 8;
                                ! Clock Cycle 8
                                ! Execute Assignment
     next;
     PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 8
     PHI2 = 10;
     DBUS<15:8> = MEABUSD;
                                ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
     DITACKN = 10;
                               ! Asserts DTACKN(Added)
     next:
                                ! Execute Pending Assignments
     ! Return To Phase 2
     T = 7
                                ! Of Clock Cycle 7
     );
                                ! Execute Impending Assignments
     next;
T = 8;
                                ! Clock Cycle 8
next;
                                ! Execute Assignment
PHI1 = 10;
                                ! Phase 2
                                ! Of Clock Cycle 8
PHI2 = hi;
EXDBUF = DBUS;
                                ! Instruction On Nata Rus
                               ! Is Placed In External Data
                                ! Bus Ruffer
                                ! Execute Pending Assignments
next;
```

```
! Clock Cycle 9
     T = 9;
     next;
                                         ! Execute Assignment
     PHI1 = bi;
                                         ! Phase 1
     PHI2 = 16:
                                         ! Of Clock Cycle 9
     IDBUS = EXDBUF;
     if EXDBUF eql 0
                                         ! Set Status Register
                                         ! Bits As Appropriate
        SRZERO = hi;
     if EXDBUF<15> eql 1
        SRNEG = hi:
     next;
                                         ! Execute Pending Assignments
     PHI1 = lo:
                                         ! Phase 2
                                         ! Of Clock Cycle 9
     PHI2 = hi;
     ASN = hi;
                                         ! Deactivate Address Strobe
     LDSN = hi;
                                         ! Deactivate Lower Data Strobe
     UDSN = hi;
                                         ! Deactivate Upper Data Strobe
     IR = PFR;
                                         ! Contents Of Prefetch Register
                                         ! Are Placed Into Instruction
                                         ! Register
                                         ! Deactivate Data Transfer(Added)
     DTACKN = hi;
                                         ! Acknowledge
     DC2J = IDBUS;
                                        ! Place Contents Of Internal
                                         ! Data Bus Into D[2]
                                         ! Execute Impending Assignments
     next;
                                         ! Reset Clock Cycle Counter
     T = 0
                                         ! JMP (A0)
.jap :=
     PHI1 = hi;
                                         ! Phase 1 Of
     PH12 = 10;
                                         ! Clock Cycle 0
     DBUS = Oxffff;
                                         ! Place Data Bus In A High Impedance
     RW = hi;
                                         ! Memory Read
                                         ! Disable Address Bus Buffer
     ALIENABLE = 10;
     DBENABLE = 10;
                                         ! Disable Data Bus Buffer
     IABUS = PC:
                                         ! Place PC On Internal Address
                                         ! Bus
                                         ! Execute Pending Assignments
     next;
                                         ! Phase 2 Of
     PHI1 = 10;
     PHI2 = hi;
                                         ! Clock Cycle 0
                                         ! Enable Address Bus Ruffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
     FCMODE = SRMODE:
                                         ! User Mode
     FCSPACE = 2:
                                         ! Accessing Program
     next;
                                         ! Execute Pending Assignments
```

```
ABUS = EXABUF;
                                ! Address Placed On Bus(Added)
                               ! Execute Pending Assignments
next;
T = 1;
                               ! Clock Cycle 1
next;
                                ! Execute Assignment
PHI1 = hi;
                               ! Phase 1 Of
PHI2 = 10:
                               ! Clock Cycle 1
ASN = lo;
                               ! Assert Address Strobe
LIISN = lo;
                               ! Assert Lower Data Strobe
                               ! Assert Upper Data Strobe
UUSN = 10;
IABUS = AEO];
                               ! Move Jump Address From A[0]
                               ! To Internal Address Buffer
                               ! Enable Data Bus
DBENABLE = hi;
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 1
PHI2 = hi;
PC = IABUS:
                               ! Place Jump Address Into Program
                               ! Counter
next;
T = 2;
                               ! Clock Cycle 2
next;
                                ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
PH12 = 10;
                               ! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
     next;
                               ! Execute Impending Assignments
     PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 2
     PHI2 = hi;
     next;
                               ! Execute Assignments
     T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
     next;
                               ! Phase 1
     PHI1 = hi;
                             ! Of Clock Cycle 3
     PHI2 = 10;
     DBUS<15:8> = MEABUSJ;
                               ! Memory Places Instruction
     DRUS<7:0> = MCABUS + 13;
                               ! On Data Bus And
     ITACKN = 10;
                                ! Asserts ITACKN(Added)
     next;
                               ! Execute fending Assignments
     ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
                               ! Execute Impending Assignments
     next;
```

```
T = 3:
                                  ! Clock Cycle 3
next;
                                  ! Execute Assignment
PHI1 = lo;
                                  ! Phase 2
                                  ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                                  ! Instruction On Data Bus
                                 ! Is Placed In External Data
                                  ! Bus Buffer
                                  ! Execute Pending Assignments
next;
T = 4;
                                 ! Clock Cycle 4
next;
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1
                                  ! Of Clock Cycle 4
PHI2 = 10;
next;
PFR = EXDBUF;
                                 ! The Contents Of The External
                                 ! Data Bus Buffer Are Placed
                                  ! In Frefetch Register
                                  ! Execute Pending Assignments
next;
PHI1 = 10:
                                 ! Phase 2
PHI2 = hii
                                 ! Of Clock Cycle 4
                                 ! Deactivate Address Strobe
ASN = hi;
LDSN = hi;
                                 ! Deactivate Lower Data Strobe
UDSN = hi;
                                 ! Neactivate Upper Data Strabe
DTACKN = hi:
                                  ! Deactivate Data Transfer
                                  ! Acknowledge(Added)
T = 5;
                                 ! Clock Cycle 5
                                  ! Execute Previous Assignment
next;
PHI1 = hi;
                                  ! Phase 1 Of
                                  ! Clock Cycle 5
PHI2 = lo;
RW = hi;
                                 ! Memory Read
                                  ! Disable Address Bus Buffer
ADENABLE = lo;
                                 ! Disable Data Bus Buffer
IRENABLE = 10;
1ABUS = FC;
                                  ! Flace PC On Internal Address
                                  ! Rus
next;
                                  ! Execute Pending Assignments
FHII = lo;
                                 ! Phase 2 Of
PHI2 = hi;
                                 ! Clock Cycle 5
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2;
                                 ! Accessing Program
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
next:
                                 ! Into External Address Buffer
ABUS = EXABUF;
                                  ! Address Floced On Bus(Added)
```

```
next;
                              ! Execute Pending Assignments
! Clock Cycle 6
next;
                              ! Execute Assignment
PHII = hi;
                              ! Phase 1 Of
                              ! Clock Cycle 6
PHI2 = 10;
ASN = 10;
                              ! Assert Address Strobe
LDSN = lo;
                              ! Assert Lower Data Strobe
UliSN = lo;
                              ! Assert Upper Data Strobe
DENABLE = hi;
                              ! Enable Data Rus
                              ! Execute Pending Assignments
next:
PHI1 = 10:
                              ! Phase 2
FHI2 = hi;
                              ! Of Clock Cycle 6
                              ! Execute Pending Assignments
next;
T = 7;
                              ! Clock Cycle 7
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1
PHI2 = lo:
                              ! Of Clock Cycle 7
while DTACKN eql hi
                              ! Wait For Memory To Place
                             ! Data On The Bus
    next;
                             ! Execute Impending Assignments
                             ! Phase 2
    PHI1 = 10;
    PHI2 = hi;
                             ! Of Clock Cycle 7
                             ! Execute Assignments
    next;
     T = 8;
                              ! Clock Cycle 8
                              ! Execute Assignment
    next;
    PHI1 = hi;
                              ! Phase 1
    PHI2 = 10;
                              ! Of Clock Cycle 8
    DBUS<15:8> = MEABUSD;
                              ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                              ! On Data Bus And
    DTACKN = 16;
                              ! Asserts DTACKN(Added)
    next;
                              ! Execute Pending Assignments
    T = 7
                             ! Return To Phase 2
                             ! Of Clock Cycle 7
     );
    next;
                             ! Execute Impending Assignments
T = 8:
                              ! Clock Cycle 8
                              ! Execute Assignment
next;
```

```
PHI1 = 10;
                                           ! Fhuse 2
     PHI2 = hi;
                                           ! Of Clock Cycle 8
     EXDRUF = DBUS;
                                           ! Instruction On Data Bus
                                          ! Is Placed In External Data
                                           ! kus kuffer
     next;
                                          ! Execute Pending Assignments
      T = 9:
                                          ! Clock Cycle 9
                                           ! Execute Assignment
     next;
                                          ! Phase 1
     PHI1 = hi;
                                           ! Of Clock Cycle 9
     PHI2 = 10:
                                           ! The Contents Of The External
     PFR = EXDBUF;
                                           ! Data Bus Buffer Are Flaced
                                          ! In Prefetch Register
                                           ! Execute Fending Assignments
     next;
     FHI1 = lo:
                                          ! Phase 2
     PHI2 = hi;
                                          ! Of Clock Cycle 9
     ASN = hi;
                                          ! Deactivate Address Strobe
                                          ! Deactivate Lower Data Strobe
     LDSN = hi;
                                          ! Deactivate Upper Data Strobe
     UDSN = hi:
     PC = PC + 2;
                                          ! Increment Program Counter
     IR = PFR;
                                          ! Place Contents Of Frefetch
                                           ! Register Into Instruction
                                          ! Register
     DTACKN = hi;
                                          ! Deactivate Data Transfer
                                           ! Acknowledge(Added)
                                          ! Execute Pending Assignments
     next;
                                          ! Reset Clock Cycle Counter
     T = 0
decode_execute_prefetch :=
                       case IR
                                           ! MOVE.W (A1).D2
                            0x3411: move
                            0x027c: andi
                                          ! AND.W #*DFFF,SR
                                           ! JMP (AO) If IR = Octal Value
                            047320: Jmp
                       esuc
main :=
    power_on_initialize;
     fetch_initial_instruction;
    while READY eql hi
          decode_execute_prefetch
     )
```

```
/ x
    MOTOROLA MC68000 MODEL OF THE MOVE.W (A1)+, D6 INSTRUCTION
/*
                                                */
/*
                                                */
/*
                                                */
/*
              Structure Declarations
                                                1/
/×
state
*/
/*
                                                */
           M68000 Programming Registers
/*
                                                */
DE0:73<31:0>,
                   ! 8 Data Registers
A[0:6]<31:0>,
                   ! 7 Address Registers
UA7<31:0>.
                   ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Pointer
PC<31:0>,
                   ! Program Counter
SR<15:0>,
                   ! Status Register
·/*
                                                */
/*
           Temporary Internal Registers
                                                */
/*
                                                */
PFR<15:0>.
                   ! Prefetch Register
IR<15:0>,
                   ! Instruction Register
FC<2:0>,
                   ! Function Code Register
EXDBUF<15:0>.
                   ! External Data Bus Buffer Register
EXABUF<23:1>,
                   ! External Address Bus Buffer Register(changed)
                   ! ALU Buffer 1
ALURUF1<31:0>.
ALUBUF2<31:0>,
                   ! ALU Buffer 2
DITEMP(15:0).
                   ! Temporary Data Storage
1/ISREG<31:0>,
                    Temporary Displacement Storage
SRTEMP<15:0>.
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>.
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPADR<31:0>,
                   ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>.
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
                   ! Temporary Vector Address Storage
VECABR<23:0>,
                   ! (Exception Processing)
```

```
HANADR<31:0>,
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
T<7:0>.
                        ! Clock Cycle Counter
FESET.
                     ! Reset Flin-Flop
HALT,
                     ! Halt Flip-Flop
Fili.
                     ! Read/Write Flip-Flop
ADENABLE,
                     ! Address Bus Buffer Enable
DIBENABLE,
                     ! Data Bus Buffer Enable
ASN.
                     ! Address Strobe Flip-Flop
LUSN.
                     ! Lower Data Strobe Flip-Flop
UDSN,
                     ! Upper Nata Strobe Flip-Flop
DITACKN.
                     ! Data Transfer Acknowledge Flip-Flop
COUT,
                     ! Carry Flip-Flop
EXCEPT,
                     ! Exception Processing Flip-Flop
READY.
                     ! Ready Flip-Flop
/ ¥
                                                              */
/*
       Model transformation modifications:
                                                              */
/×
                                                              */
/ x
           1) CDL decoder structure nonexistent in ISF' and un-
                                                              */
/*
       necessary for model. Eliminated.
                                                              */
/*
           Multi-phase clock structure nonexistent in ISP'.
                                                              */
/*
       Operations on registers will provide its equivalent.
                                                              */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
/*
       register will provide its equivalent.
                                                              */
/*

 The declared bus structures are modeled with registers */

/¥
       without loss of model accurracy. This done to maintain model
                                                             */
/x
       equivalency and simplicity.
                                                              */
/x
           5) The memory word length was reduced from 16 to 8 bit
                                                             */
/x
       words to coincide with the ECR's 32-Kbyte memory, to agree with*/
/*
       their PC incrementation, and to enable the use of existing
                                                             */
/*
       MC68000 assembler and linker/loader models. The memory was
                                                             */
       also reduced from 8 hwords to 32 Kbytes.
/±
                                                             */
/*
                                                             */
IABUS<31:0>,
                        ! Internal Address Bus
IDBUS<31:0>,
                        ! Internal Data Bus
twait(7:0),
                        ! Wait Cycle Counter
SWITCH,
                     ! Power Switch
                     ! Phase 1 Of Two-Phase Clock
PHI1,
PHI2;
                     ! Phase 2 Of Two-Phase Clock
port
/*
                                                              */
                                                              */
/*
             External Address and Data Bus
/*
DBUS<15:0>,
                        ! External Data Bus
```

```
ABUS<23:1>;
                     ! External Address Bus(changed)
format
/*
                                                        */
/*
                                                        */
                Register Subfields
/*
                                                        */
PCADDR
         = PC<23:0>,
                      ! Program Counter Address Field
SRTRACE
         = SR<15>.
                      ! Trace Bit
SRMODE
         = S8<13>.
                      ! Mode Selection Bit
SRCARRY
         = SR<0>,
                      ! Carry Rit
SROVER
         = SR<1>,
                      ! Overflow Bit
                      ! Zero Bit
SRZERO
         = SR<2>
SKNEG
         = SR<3>,
                      ! Negative Bit
         = SR<4>,
SREX
                      ! Extend Bit
SRMASK
         = SR<10:8>,
                       Interrupt Mask
         = FC<1:0>,
FICSPIACE
                      ! Memory Access Address Space
FCMODE
         = FC<2>.
                      ! User/Supervisor Mode Rit
FCLOW
         = PC<15:0>.
                      ! PC Low Word
         = PC<31:16>,
PCHI
                      ! PC High Word
LIOLWORD
         = D[0]<15:0>.
                      ! DEOJ Low Word
DILWORD
         = D[1]<15:0>,
                      ! D[1] Low Word
I/2LWORD
         = DE23<15:0>,
                      ! DE23 Low Word
         = DE33<15:0>,
D3LWORD
                      ! DE31 Low Word
I/4LWORD
         = D[4](15:0),
                      ! DE43 Low Word
DSLWORD
                      ! D[5] Low Word
         = D(5)(15:0),
DIGLWORD
         = DE63<15:0>,
                      ! II[6] Low Word
D7LWORD
         = DE73<15:0>,
                      ! BE73 Low Word
DISREGHWORD = DISREG(31:16), | DISREG High Word
DISREGLWORD = DISREG(15:0), ! DISREG Low Word
HANATIRLOW
         = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANAUR<31:16>,! HANAUR High Word
TEMPADRLOW = TEMPADR<15:0>.! TEMPADR Low Word
         = TEMPADR<31:16>;! TEMPADR High Word
TEMPAURHI
Memory
/*
                                                        */
/*
                16K 16-Bit Word Internal Memory
                                                        */
/#
                                                        */
ME0:327673<7:0>:
macro
/*
/*
                                                        */
               Lugic Level Macros
```

```
/ ¥
= 0 %,
10
     = 1 %,
hi
off
     = 0 %.
     = 1 %,
On
clear = 0 &;
x/
/*
                                                            */
/* Fower On and Initialization. This process was not modeled but is
   added to initialize signals and registers.
                                                            */
                                                            */
/*
power_on_initialize :=
      SWITCH = on;
                                  ! Turn Fower On
      next;
                                  ! Execute Assignment
      READY = lo;
                                  ! System Not Ready
                                  ! Assert Reset For
      RESET = lo;
                                  ! 100 Miliseconds(Active Low)
       delay(100);
      RESET = hi;
                                  ! Deactivate Reset
                                  ! Execute Pending Assignments
      next;
                                  ! Initialize Address Strobe
      ASN = hi;
                                  ! Initialize Lower Data Strobe
      LDSN = hi;
                                  ! Initialize Upper Data Strobe
      UDSN = hi:
                                  ! Initialize Data Transfer Acknowledge
      INTACKN = hi;
                                  ! Initialize Read/Write(Read On High)
      RW = hi;
      DROS = Oxffff;
                                  ! Place Data Bus In High Impedance State
      MEOx100cl = Oxff;
                                  ! Place Memory Locations Following The
      ME0 \times 100dJ = 0 \times ff;
                                    ! JMP Instruction In A High State
      HALT = hi;
                                  ! Initialize Halt Flip-Flop(Active
                                  ! LOW)
       T = 0;
                                  ! Initialize Clock Cycle Counter
      READY = hi;
                                  ! System Ready
       */
       /*
                                                            */
       /*
            Routine Initialization Fer Hamby and Guillory
                                                            */
       /*
       ! Initialize Memory Location
      ME0x20000 = 0x55;
      ME0x20013 = 0x55;
                                  ! 2000 Hex To 5555 Hex
       AE03 = 0 \times 1004;
                                  ! Flace Hex 1004 Into ACOJ
       A[1] = 0x2000;
                                  ! Store Data At Hex 2000
                                  ! DC23 Will Reset AC13
       D[23 = 0 \times 2000;
      ME0\times2002] = 0\times aa;
                                  ! Nata Will Also Be Moved
       M[0x2003] = 0xaa;
                                  ! Place Hex 1000 Into Program Counter
       F'C = 0x1000;
                                  ! Execute Assignments
       next
```

```
/*
/x
  Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                         x/
  to retrieve modeled instructions for simulation and analysis. It
/*
                                                         */
  was fashsioned from the Read Cycle described by Hamby and Guillory */
/* on page VI-15 of their thesis.
/*
fetch_initial_instruction :=
    ! Phase 1 Of
    PHI1 = hi;
    PHI2 = 1o:
                                   ! Clock Cycle 0
    RW = hi:
                                   ! Memory Read
    ADENABLE = lo:
                                   ! Disable Address Bus Buffer
    DRENABLE = 10;
                                   ! Disable Data Bus Buffer
    IABUS = PC;
                                   ! Place PC On Internal Address
    next;
                                   ! Execute Fending Assignments
    PHI1 = lo;
                                   ! Phase 2 Of
                                   ! Clock Cycle 0
    PHI2 = hi;
                                   ! Enable Address Bus Buffer
    ADENABLE = hi;
    EXABUF = IABUS;
                                   ! Gate Internal Address Bus
                                   ! Into External Address Buffer
                                   ! User Mode
    FCMODE = SRMODE;
    FCSFACE = 2;
                                   ! Accessing Program
    next;
                                   ! Execute Impending Assignments
    ABUS = EXABUF;
                                   ! Address Placed On Bus(Added)
    next;
                                   ! Execute Pending Assignments
    T = 1;
                                   ! Clock Cycle 1
    next;
                                   ! Execute Assignment
    PHI1 = hi;
                                   ! Phase 1 Of
                                   ! Clock Cycle 1
    PHI2 = 10;
    ASN = 10;
                                   ! Assert Address Strobe
    LISN = lo;
                                   ! Assert Lower Data Strobe
    UDSN = 10:
                                   ! Assert Upper Data Strobe
    DIBENABLE = ha;
                                   ! Enable Data Bus
                                   ! Execute Pending Assignments
    next:
    FHI1 = lo:
                                   ! Phase 2
    PHI2 = hi;
                                   ! Of Clock Cycle 1
    next:
                                   ! Execute Pending Assignments
    T = 2;
                                   ! Clock Cycle 2
    next;
                                   ! Execute Assignment
```

```
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
PHI2 = 10;
                               ! Wait For Memory To Place
while BTACKN eql ha
                               ! Nata On The Bus
                               ! Execute Impending Assignments
     next;
     PHI1 = lo;
                               ! Phase 2
     PHI2 = hi;
                               ! Of Clock Cycle 2
                               ! Execute Assignments
     next;
     ! Clock Cycle 3
     T = 3;
     next;
                               ! Execute Assignment
                               ! Phase 1
     PHI1 = hi;
                               ! Of Clock Cycle 3
     PHI2 = 10;
                               ! Memory Places Instruction
     DBUS<15:8> = MEABUSD;
                               ! On Data Bus And
     DBUS<7:0> = MEABUS + 13;
     DTACKN = 10;
                               ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
     next;
     ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
                               ! Execute Impending Assignments
     next;
! Clock Cycle 3
T = 3;
                               ! Execute Assignment
next;
PHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
                               ! Instruction On Data Bus
EXDBUF = DBUS;
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
! Clock Cycle 4
T = 4:
                               ! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi;
PHI2 = 10;
                               ! Of Clock Cycle 4
                               ! The Contents Of The External
PFR = EXDBUF;
                                ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
                               ! Execute Fending Assignments
next;
                               ! Phase 2
PHI1 = lo;
                               ! Of Clock Cycle 4
PHI2 = bi;
                               ! Deactivate Address Strobe
ASN = h1;
```

CONTRACTOR OF THE CONTRACTOR O

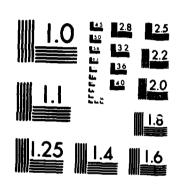
```
LDSN = hi;
                                         ! Neactivate Lower Nata Strobe
                                         ! Deactivate Upper Data Strobe
     UDSN = hi;
     IR = PFR:
                                         ! Contents Of Prefetch Register
                                         ! Are Placed Into Instruction
                                         ! Register
     DTACKN = hi;
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge.
     FC = FC + 4;
                                         ! Increment Program Counter
                                         ! Execute Pending Assignments
     next;
                                         ! Reset Clock Cycle Counter
     T = 0
                                         ! MOVE.L D2,A1
movel :=
     ! Phase 1 Of
     PHI1 = hi;
                                         ! Clock Cycle 0
     PHI2 = 10;
     AMENABLE = 10;
                                         ! Disable Address Bus
                                         ! Disable Data Bus
     DRENABLE = 10:
                                         ! Place Nata Bus In High Impedance
     DBUS = 0xffff;
                                         ! Memory Read
     f:W = hi:
                                         ! Place PC On Internal Address
     IARUS = PC;
                                         ! Bus
     IDBUS = DC23;
                                      ! Place Data From DC23 Onto
                                         ! Internal Data Bus
                                         ! Execute Pending Assignments
     next;
     PHI1 = 10;
                                         ! Phase 2 Of
     PHI2 = hi;
                                         ! Clock Cycle 0
     ADENABLE = hi;
                                         ! Enable Address Bus Buffer
                                         ! Gate Internal Address Bus
     EXABUF = IABUS;
                                         ! Into External Address Buffer
     FCMUDE = SRMODE;
                                         ! User Mode
     FCSFACE = 2;
                                         ! Accessing Program
     SRCARRY = lo;
                                         ! Clear Status Register Carry Bit
     SROVER = lo;
                                         ! Clear Status Register Overflow Bit
                                         ! Clear Status Register Zero Rit
     SRZERO = lo;
     SRNEG = 10;
                                         ! Clear Status Register Negative Bit
     AC1] = IDBUS;
                                      ! Place Data From Internal Data Bus
                                         ! Into A[1]
                                         ! Execute Impending Assignments
     next;
                                         ! Address Placed On Bus(Added)
     ABUS = EXABUF;
     next;
                                         ! Execute Pending Assignments
     T = 1;
                                         ! Clock Cycle 1
                                         ! Execute Assignment
     next;
                                         ! Phase 1 Of
     PHI1 = hi:
     PHI2 = 10:
                                         ! Clock Cycle 1
                                         ! Assert Address Strobe
     ASN = lo;
```

```
LUSN = lo:
                                ! Assert Lower Data Strobe
UI(SN = 10;
                                ! Assert Upper Data Strobe
                                ! Enable Data Bus
DBENABLE = hi;
                             ! Set Status Register Zero Bit
if A[1] eql 0
                                ! If Moved Data Is Zero
   SRZERO = hi;
next;
                                ! Execute Pending Assignments
PHI1 = 10;
                                ! Phase 2
                                ! Of Clock Cycle 1
PHI2 = hi;
if AE13<31>
                                ! Set Status Register Negative
   SRNEG = hi;
                                ! Bit If Moved Data Is Negative
                                ! Execute Pending Assignments
next:
! Clock Cycle 2
next;
                                ! Execute Assignment
PHI1 = hi:
                                ! Phase 1
                                ! Of Clock Cycle 2
PHI2 = 10;
while DTACKN eql hi
                                ! Wait For Memory To Place
                                ! Data On The Bus
     next;
                                ! Execute Impending Assignments
     PHI1 = 10;
                                ! Phase 2
                                ! Of Clock Cycle 2
     PHI2 = hi:
                                ! Execute Assignments
     next;
     T = 3;
                                ! Clock Cycle 3
                                ! Execute Assignment
     next;
     PHI1 = hi;
                               ! Phase 1
     PHI2 = 10:
                                ! Of Clock Cycle 3
     DBUS<15:8> = M[ABUS];
                                ! Memory Places Instruction
                                ! On Data Bus And
     DBUS<7:0> = MEABUS + 13;
     DTACKN = lo;
                                ! Asserts DTACKN(Added)
                                ! Execute Pending Assignments
     next;
     ! Return To Phase 2
     T = 2
                                ! Of Clock Cycle 2
     );
     next;
                                ! Execute Impending Assignments
T = 3:
                                ! Clock Cycle 3
next;
                                ! Execute Assignment
f'HI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 3
EXDRUF = DBUS;
                                ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Bus Buffer
```

```
! Execute Pending Assignments
     next;
     T = 4;
                                         ! Clock Cycle 4
                                         ! Execute Assignment
     next;
                                         ! Phase 1
     PHI1 = hi;
                                         ! Of Clock Cycle 4
     PHI2 = 10;
     PFR = EXDBUF;
                                         ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Prefetch Register
                                         ! Execute Pending Assignments
     next;
     PHI1 = 10;
                                         ! Phase 2
     PHI2 = hi;
                                         ! Of Clock Cycle 4
                                         ! Deactivate Address Strobe
     ASN = hi;
                                         ! Neactivate Lower Nata Strobe
     LDSN = hi:
                                         ! Deactivate Upper Data Strobe
     UDSN = hi;
                                         ! Contents Of Prefetch Register
     IR = PFR;
                                         ! Are Flaced Into Instruction
                                         ! Register
     DTACKN = hi;
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
     PC = PC + 2;
                                         ! Increment Program Counter
     next:
                                         ! Execute Impending Assignments
     T = 0
                                         ! Reset Clock Cycle Counter
     )
andi :=
                                         ! AND.W #$DFFF,SR
                                         ! Effect Of Instruction
    SRMODE = 10;
    IR<15:8> = MCPC3;
                                         ! Prefetch Next Instruction
    IR<7:0> = MCPC + 13;
                                         ! Is To Set Status Register
    next;
                                            ! Increment Program Counter
       PC = PC + 2;
    T = 5;
                                         ! Supervisor Rit To User
                                         ! Mode
    next;
    T = 0
                                         ! Requires 6 Clock Cycles
    )
                                         ! MOVE.W (A1)+,D6
move :=
     PHI1 = hi;
                                         ! Phase 1 Of
                                         ! Clock Cycle 0
     PHI2 = 10;
     DBUS = Oxffff;
                                         ! Place Data Bus In High Impedance
     RW = hi;
                                         ! Memory Read
     ADENABLE = 10;
                                         ! Disable Address Bus Buffer
     ABUS = 0xffffff;
                                        ! Address Rus High Impedanced
     DIBENABLE = 10;
                                        ! Disable Data Rus Buffer
```

(

THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 HICROP. (U) RIR FORCE INST OF TECH MRIGHT-PATTERSON AFB OH SCHOOL OF ENGI. C A BAXLEY DEC 84 AFIT/GCS/ENG/840-2-VOL-2 F/G 9/2 MO-R164 257 3/3 UNCLASSIFIED NL



THE STATE OF THE S

MICROCOPY RESOLUTION TEST CHART

```
IABUS<31:1> = PC<31:1>;
                                ! Place PC On Internal Address
                                 Rus
                                ! Execute Pending Assignments
next;
PHI1 = lo;
                                ! Phase 2 Of
PHI2 = hi:
                                ! Clock Cycle 0
ADENABLE = hi:
                               ! Enable Address Bus Buffer
EXABUF = IABUS(23:1);
                                ! Gate Internal Address Bus
                                ! Into External Address Buffer
FCMODE = SRMODE;
                                ! User Mode
FCSFACE = 2;
                                ! Accessing Program
ABUS = IABUS<23:1>;
                               ! Flace FC On Address Bus
next:
                               ! Execute Impending Assignments
! Clock Cycle 1
T = 1;
                                ! Execute Assignment
next:
                                ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                                ! Clock Cycle 1
                                ! Assert Address Strobe
ASN = 10;
LIISN = 10:
                                ! Assert Lower Data Strobe
                               ! Assert Upper Data Strobe
UI(SN = 10;
DBENABLE = hi;
                               ! Enable Data Bus
                                ! Execute Pending Assignments
next;
                               ! Phase 2
PHI1 = lo;
                                ! Of Clock Cycle 1
FHI2 = bi;
                                ! Execute Pending Assignments
next:
T = 2;
                                ! Clock Cycle 2
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 2
FHI2 = 10;
                                ! Wait For Memory To Flace
while DTmCKN eql hi
                                ! Nata On The Rus
                                ! Execute Impending Assignments
     next;
     I'HI1 = 10;
                                ! Phase 2
     PHI2 = hi;
                                ! Of Clock Cycle 2
     next;
                                ! Execute Assignments
     ! Clock Cycle 3
     T = 3:
     next;
                                ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 3
     PHI2 = 10;
     DBUS<15:8> = MCABUSD;
                               ! Memory Fluces Instruction
     DBUS(7:0) = MEABUS + 13;
                               ! On Data Bus And
                                ! Asserts DTACKN(Added)
     DITACKN = 10;
```

```
! Execute Fending Assignments
     next;
     ! Return To Phase 2
                               ! Of Clock Eyele 2
     ):
                               ! Execute Impending Assignments
     next;
T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
next;
                               ! Phase 2
FHI1 = 10;
                               ! Of Clock Cycle 3
PHI2 = hi;
EXPROF = DRUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
next;
                               ! Execute Pending Assignments
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PH11 = hi:
                               ! Phase 1
                               ! Of Clock Cycle 4
PHI2 = 10:
                               ! The Contents Of The External
PFR = EXDBUF;
                               ! Data Bus Buffer Are Placed
                               ! In Frefetch Register
                               ! Execute Pending Assignments
next;
FHI1 = 10;
                               ! Phase 2
FHI2 = hi;
                               ! Of Clock Cycle 4
                               ! Deactivate Address Strobe
ASN = hi;
LDSN = hi;
                               ! Deactivate Lower Data Strobe
                               ! Deactivate Upper Data Strobe
UDSN = hi;
PC = PC + 2;
                               ! Increment Program Counter
                               ! Are Placed Into Instruction
                               ! Register
IJTACKN = hi:
                               ! Deactivate Data Transfer(Added)
                                ! Acknowledge
next:
T = 5;
                               ! Clock Cycle 5
                               ! Execute Previous Assignment
next;
                               ! Phase 1 Of
PHI1 = hi:
                               ! Clock Cycle 5
FHI2 = 10;
                              ! Flace Data Bus In High Impedance
INBUS = Oxffff;
RW = hi
                               ! Memory Read
ADENABLE = 10;
                               ! Disable Address Bus Buffer
ARUS = 0xffffff;
                               ! Address Rus High Impedanced
DRENABLE = 10;
                               ! Disable Data Bus Buffer
```

```
IABUS = A[1];
                                ! Place A[1] On Internal Address
                                ! Rus
                                ! Execute Pending Assignments
next;
                                ! Phase 2 Of
FHII = 10:
PHI2 = hi:
                                ! Clock Cycle 5
ADENABLE = hi;
                               ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;
                               ! Gate Internal Address Bus
                                ! Into External Address Buffer
FCMUDE = SRMODE:
                                ! User Mode
FCSPACE = 1;
                                ! Accessing Data
SRCARRY = 10;
                               ! Clear Status Register Carry Bit
                               ! Clear Status Register Overflow Bit
SROVER = 10;
                                ! Clear Status Register Zero Bit
SRZERO = lo;
SRNEG = 10;
                                ! Clear Status Register Negative Bit
ABUS = IABUS<23:1>;
                               ! Place PC On Address Rus (Added)
                                ! Execute Impending Assignments
! Clock Cycle 6
T = 6:
                                ! Execute Assignment
next;
                                ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                                ! Clock Cycle 6
ASN = lo;
                                ! Assert Address Strobe
                                ! Assert Lower Data Strobe
LUSN = 10;
                               ! Assert Upper Data Strobe
UDSN = lo:
                                ! Enable Data Bus
DBENABLE = hi;
                                ! Execute Pending Assignments
next;
PHI1 = lo;
                                ! Phase 2
FHI2 = hi;
                                ! Of Clock Cycle 6
                                ! Execute Pending Assignments
next;
! Clock Cycle 7
T = 7:
                                ! Execute Assignment
next:
                               ! Phase 1
Filli = hi:
                               ! Of Clock Cycle 7
PHI2 = 10:
                                ! Wait For Memory To Place
while DTACKN eql hi
                                ! Data On The Rus
                                ! Execute Impending Assignments
     next;
     f'HI1 = lo;
                               ! Phase 2
                                ! Of Clock Cycle 7
     PHI2 = hi;
                                ! Execute Assignments
     next:
     ! Clock Cycle 8
     T = 8:
                                 ! Execute Assignment
     next;
     FH11 = hi;
                               ! Phase 1
```

```
! Of Clock Cycle 8
     PHI2 = 10:
     DRUS<15:8> = MEARUSD;
                                  ! Memory Places Instruction
     DBUS (7:0) = MCABUS + 13;
                                 ! On Data Bus And
                                  ! Asserts INTACKN(Added)
     INTACKN = 10:
                                  ! Execute Pending Assignments
     next:
     T = 7
                                  ! Return To Phase 2
                                  ! Of Clock Cycle 7
     );
     next;
                                 ! Execute Impending Assignments
T \approx 8;
                                  ! Clock Cycle 8
next;
                                  ! Execute Assignment
f'HI1 = 16:
                                  ! Phase 2
PHI2 = hi:
                                  ! Of Clock Cycle 8
EXDRUF = DRUS;
                                 ! Instruction On Data Bus
                                  ! Is Placed In External Data
                                 ! Bus Buffer
A[1] = A[1] + 2;
                                  ! Increment A[1]
                                 ! Execute Fending Assignments
next;
T = 9;
                                  ! Clock Cycle 9
next;
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1
PHI2 = lo;
                                  ! Of Clock Cycle 9
IDBUS = EXDBUF;
if EXDBUF eql 0
                                  ! Set Status Register
  SRZERO = hi;
                                  ! Bits As Appropriate
if EXDBUF<15> eal 1
  SKNEG = hi;
next;
                                  ! Execute Fending Assignments
PHI1 = 10;
                                  ! Phase 2
PHI2 = hi:
                                  ! Of Clock Cycle 9
if IR eq1 0x3c19
                                  ! Place Contents Of Internal
  INTER = INBUS
                                  ! Data kas Into DE63/DE73
  NI73 = IDBUS:
ASN = hi:
                                  ! Deactivate Address Strobe
LDSN = hi;
                                  ! Deactivate Lower Data Strobe
UDSN = hi;
                                  ! Deactivate Upper Data Strobe
IR = PFR;
                                  ! Contents Of Prefetch Register
                                  ! Are Flaced Into Instruction
                                  ! Register
LITACKN = hi;
                                  ! Deactivate Data Transfer(Added)
                                  ! Acknowledge
                                  ! Execute Impending Assignments
next;
T = 0
                                  ! Reset Clock Cycle Counter
```

```
=: بىئار
                                          1 JMF (A0)
      PHI1 = hi:
                                          ! Phase 1 Of
      PHI2 = 10:
                                          ! Clock Cycle 0
      DBUS = 0xffff;
                                          ! Place Dato Bus In A High Impedance
      RW = hi;
                                          ! Memory Read
      ADENABLE = 10;
                                          ! Disable Address Bus Buffer
      DIBENABLE = 10;
                                          ! bisable Data Bus Buffer
      IARUS = PC;
                                          ! Place PC On Internal Address
      next;
                                          ! Execute Pending Assignments
      PHI1 = 10;
                                          ! Phase 2 Of
      PHI2 = hi;
                                          ! Clock Cycle 0
                                          ! Enable Address Bus Buffer
      AVENABLE = hi;
      EXABUF = IABUS;
                                          ! Gate Internal Address Bus
                                         ! Into External Address Buffer
      FCMODE = SRMODE;
                                          ! User Mode
      FCSFACE = 2;
                                          ! Accessing Program
                                          ! Execute Pending Assignments
      next;
      ABUS = EXABUF;
                                          ! Address Flaced On Rus(Added)
                                          ! Execute Pending Assignments
      next:
      T = 1;
                                          ! Clock Cycle 1
      next;
                                          ! Execute Assignment
      PHI1 = hi;
                                          ! Phase 1 Of
      PHI2 = 10;
                                          ! Clock Cycle 1
      ASN = 10;
                                          ! Assert Address Strobe
      LISN = lo:
                                          ! Assert Lower Data Strobe
      UDSN = lo:
                                          ! Assert Upper Data Strobe
      IABUS = ACOJ;
                                          ! Move Jump Address From ACOJ
                                          ! To Internal Address Buffer
      DBENABLE = hi;
                                          ! Enable Data Rus
                                          ! Execute Pending Assignments
      next;
      PHI1 = 10;
                                          ! Phase 2
      PHI2 = hi;
                                          ! Of Clock Cycle 1
      PC = IABUS;
                                          ! Place Jump Address Into Program
                                          ! Counter
      next;
      T = 2:
                                         ! Clock Cycle 2
      next;
                                          ! Execute Assignment
      FHI1 = hi;
                                          ! Phase 1
```

```
PHI2 = 1o;
                               ! Of Clock Cycle 2
                               ! Wait For Memory To Flace
while DTACKN eal hi
                               ! Nata On The Rus
                               ! Execute Impending Assignments
     next;
     PHI1 = 10;
                               ! Phase 2
     PHI2 = hi;
                               ! Of Clock Cycle 2
     next;
                               ! Execute Assignments
     ! Clock Cycle 3
     T = 3;
     next;
                               ! Execute Assignment
                               ! Phase 1
     fHI1 = hi;
     PHI2 = 10;
                               ! Of Clock Cycle 3
                               ! Memory Places Instruction
     DRUS<15:8> = MEARUSD;
     DBUS<7:0> = MEABUS + 1];
                               ! On Data Bus And
     DITACKN = 10;
                               ! Asserts ITACKN(Added)
     next;
                               ! Execute Pending Assignments
     T = 2
                               ! Return To Phase 2
                                ! Of Clock Cycle 2
     );
     next;
                               ! Execute Impending Assignments
T = 3;
                             ! Clock Cycle 3
next;
                               ! Execute Assignment
FHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 3
EXDBUF = DRUS;
                               ! Instruction On Data Bus
                               ! Is Flaced In External Data
                                ! Rus Buffer
next;
                               ! Execute Pending Assignments
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 4
next;
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Nata Bus Ruffer Are Flaced
                                ! In Prefetch Register
next;
                               ! Execute Pending Assignments
FHI1 = 10;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 4
ASN = hi;
                               ! Deactivate Address Strobe
                               ! Deactivate Lower Data Strobe
LUSN = hi;
```

```
UDSN = hi;
                                 ! Deactivate Upper Data Strobe
DTACKN = hi;
                                 ! Deactivate Data Transfer
                                 ! Acknowledge(Added)
next:
! Clock Cycle 5
                                 ! Execute Previous Assignment
next;
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 5
RW = hi;
                                 ! Memory Read
ADENABLE = 10;
                                 ! Disable Address Bus Buffer
DEENnBLE = 10;
                                 ! Disable Data Bus Buffer
IABUS = FC;
                                 ! Place PC On Internal Address
                                 ! Bus
next;
                                 ! Execute Pending Assignments
                                 ! Phase 2 Of
PHI1 = lo;
                                 ! Clock Cycle 5
PHI2 = hi;
                                 ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                 ! User Mode
                                 ! Accessing Program
FCSPACE = 2;
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
                                 ! Into External Address Buffer
next;
                                 ! Address Flaced On Rus(Added)
ABUS = EXABUF:
                                 ! Execute Pending Assignments
next;
T = 6:
                                 ! Clock Cycle 6
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 6
                                 ! Assert Address Strobe
ASN = 10;
LISN = lo;
                                 ! Assert Lower Data Strobe
                                 ! Assert Upper Data Strobe
UDSN = 10;
DEENABLE = hi;
                                 ! Enable Nata Rus
next;
                                 ! Execute Pending Assignments
                                 ! Phase 2
F'HI1 = lo;
fHI2 = hi;
                                 ! Of Clock Cycle 6
next:
                                 ! Execute Pending Assignments
T = 7;
                                 ! Clock Cycle 7
                                 ! Execute Assignment
next;
PHI1 = hi;
                                 ! Phase 1
                                 ! Of Clock Cycle 7
PHI2 = 10;
                                 ! Wait For Memory To Place
while DTACKN eql hi
                                 ! Data On The Bus
                                 ! Execute Impending Assignments
     next;
```

```
! Phase 2
     PHI1 = 10;
     PHI2 = hi;
                                ! Of Clock Cycle 7
                                ! Execute Assignments
     next;
     ! Clock Cycle 8
     next;
                                ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
     FHI2 = 10;
                                ! Of Clock Cycle 8
     DBUS(15:8> = MEABUS];
                                ! Memory Places Instruction
     BBUS <7:0> = MEABUS + 13;
                               ! On Data Rus And
                                ! Asserts DTACKN(Added)
     DITACKN = 10;
                                ! Execute Fending Assignments
     next;
     T = 7
                               ! Return To Phase 2
                               ! Of Clock Cycle 7
     );
     next;
                               ! Execute Impending Assignments
T = 8:
                                ! Clock Cycle 8
next;
                                ! Execute Assignment
PHI1 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 8
EXDBUF = DBUS:
                                ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Rus Ruffer
next;
                                ! Execute Pending Assignments
T = 9;
                                ! Clock Cycle 9
next;
                                ! Execute Assignment
FHI1 = hi;
                                ! Phase 1
FHI2 = 10;
                                ! Of Clock Cycle 9
PFR = EXIMUF:
                                ! The Contents Of The External
                                ! Data Bus Buffer Are Placed
                                ! In Prefetch Register
                                ! Execute Pending Assignments
next;
PHI1 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 9
                                ! Deactivate Address Strobe
ASN = hi;
                                ! Deactivate Lower Data Strobe
LDSN = hi:
UDSN = hi;
                                ! Deactivate Upper Data Strobe
                                ! Increment Program Counter
PC = PC + 2;
IR = PFR;
                                ! Place Contents Of Prefetch
                                ! Register Into Instruction
                                ! Register
DTACKN = hi;
                                ! Deactivate Data Transfer
```

```
! Acknowledge(Added)
      next;
                                             ! Execute Pending Assignments
      T = 0
                                             ! Reset Clock Cycle Counter
decode_execute_prefetch :=
                        case Ik
                                    0x2242: movel
                                                    ! MOVE.L D2,A1
                             0x3c19,0x3e19; move
                                                  ! MOVE.W (A1)+,D6
                                    0x027c: andi
                                                  ! AND.W #4DFFF,SR
                                    047320: Jap
                                                   ! JMP (AO)
                        esac
main :=
     power_on_initiulize;
     fetch_initial_instruction;
     while READY eql hi
           decode_execute_prefetch
     )
```

· • •

```
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W -(A1), D4 INSTRUCTION
                                                */
/*
                                                */
/*
/*
              Structure Declarations
                                                1/
/*
                                                */
state
/*
/*
                                                */
           M68000 Programming Registers
/*
                                                */
DE0:73<31:0>,
                   ! 8 Nata Registers
                   ! 7 Address Registers
A[0:6]<31:0>,
UA7<31:0>.
                   ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Pointer
                   ! Program Counter
PC<31:0>,
SR<15:0>,
                   ! Status Register
*/
/¥
/*
                                                1/
           Temporary Internal Registers
PFR<15:0>,
                   ! Frefetch Register
IR<15:0>,
                   ! Instruction Register
FC<2:0>,
                   ! Function Code Register
                   ! External Data Bus Buffer Register
EXDBUF<15:0>,
EXABUF<23:1>,
                   ! External Address Bus Buffer Register(changed)
ALURUF1<31:0>.
                   ! ALU Buffer 1
                   ! ALU Buffer 2
ALUBUF2<31:0>,
DITEMP<15:0>.
                   ! Temporary Data Storage
I/1SREG<31:0>.
                   ! Temporary Displacement Storage
SKTEMP<15:0>,
                   ! Temporary Status Register Storage
                   ! (Exception Frocessing)
IRTEMP<15:0>.
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
                   ! Temporary Cycle Address Storage
TEMPADR<31:0/.
                   ! (Exception Processing)
ACTYPE (15:0),
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>,
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:0>,
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
T<7:0>,
                        ! Clock Cycle Counter
RESET,
                      ! Reset Flip-Flop
HALT,
                      ! Halt Flip-Flop
F.W.
                      ! Read/Write Flip-Flop
ADENABLE,
                      ! Adoress Bus Buffer Enable
                      ! Data Bus Buffer Enable
DIBENABLE,
ASN.
                      ! Address Strobe Flip-Flop
LISN,
                      ! Lower Nata Strobe Flip-Flop
UDSN,
                      ! Upper Data Strobe Flip-Flop
DITACKN.
                      ! Nata Transfer Acknowledge Flip-Flop
COUT,
                      ! Carry Flip-Flop
EXCEPT,
                      ! Exception Processing Flip-Flop
READY,
                      ! Ready Flip-Flop
/x
                                                              1/
/x
                                                              x/
       model transformation modifications:
/*
                                                              */
/*
           1) CDL decoder structure nonexistent in ISP' and un-
                                                              */
11
       necessary for model. Eliminated.
                                                              */
/*
           Multi-phase clock structure nonexistent in ISP'.
                                                              */
/*
       Operations on registers will provide its equivalent.
                                                              */
/*
           3) Switch structure nonexistent in TSP'. Operation on a
                                                              */
/*
       register will provide its equivalent.
                                                              */
/*
           4) The declared bus structures are modeled with registers */
/*
       without loss of model accurracy. This done to maintain model
                                                              X/
11
       equivalency and simplicity.
                                                              */
/*
           5) The memory word length was reduced from 16 to 8 bit
                                                              */
/x
       words to coincide with the ECR's 32-Kbyte memory, to agree with*/
/×
       their PC incrementation, and to enable the use of existing
                                                              */
/*
       MCC8000 assembler and linker/loader models. The memory was
                                                              */
/*
       also reduced from 8 Mwords to 32 Kbytes.
                                                              */
/*
                                                              */
! Internal Address Rus
1ABUS(31:0),
IDBUS(31:0>,
                        ! Internal Data Rus
twait<4:0>,
                        ! Wait State Counter
SWITCH,
                      ! Fower Switch
PHI1,
                      ! Phase 1 Of Two-Phase Clock
PHI2;
                      ! Phase 2 Of Two-Phase Clock
port
/*
                                                              */
/*
              External Address and Data Bus
                                                              */
/*
                                                              */
DBUS<15:0>,
                        ! External Data Rus
```

```
ABUS<23:10;
                      ! External Address Bus(changed)
format
*/
/*
                 Register Subfields
                                                         */
                                                         */
1*
PCAUDR
                      ! Program Counter Address Field
         = FC(23:0),
SRTRACE
         = SR<15>,
                      ! Trace Bit
SKMODE
         = SR<13>,
                      ! Mode Selection Rit
SRCARRY
         = SR<0/
                      ! Curry Bit
         = SR<1>,
SROVER
                      ! Overflow Bit
SRZERO
         = SR(2),
                       Zero Bit
         = SR<3>,
SRNEG
                      ! Negative Bit
SREX
         = SR<4>.
                      ! Extend Bit
SHMASK
         = SR<10:8>,
                      ! Interrupt Mask
         = FC<1:0>,
FCSFACE
                      ! Memory Access Address Space
FCMODE
         = FC(2),
                      ! User/Supervisor Mode Bit
                      ! PC Low Word
FCLOW
         = PC<15:0>,
PCH1
                      ! PC High Word
         = PC<31:16>,
DOLWORD
         = I([0]<15:0),
                      ! IIIO3 Low Word
DILWORD
         = DE13<15:0>,
                      ! [[1] Low Word
I/2LWORD
         = DC23<15:0>,
                      ! DE23 Low Word
DBLWORD
                      ! DE33 Low Word
         = DE33<15:0>,
D4LWORD
                      ! 1/143 Low Word
         = R[43<15:0>,
DSLWORD
                      ! DES] Low Word
         = 1653(15:6),
DIGLWORD
                      ! II[6] Low Word
         = DE63<15:0>.
                      ! DE73 Low Word
D7LWORD
         = DE73<15:0>,
DISREGHWORD = DISRES<31:16>,! DISREG High Word
DISREGLWORD = DISREG(15:0), ! DISREG Low Word
HANADELOW
         = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
         = TEMPADR<31:16>;! TEMPADR High Word
TEMPADRHI
MEMORY
*/
/*
/*
                 16K 16-Bit Word Internal Memory
                                                         */
/*
                                                         */
ME0:327673 (7:0);
mac no
/*
                                                         */
/*
                                                         x/
                Logic Level Macros
```

(1) はないできる。第一次できるのでは、大きなのである。例できるののは特になるなるながではなるのでは、はないできない。

```
= 0 1,
    = 1 %,
h1
    = 0 %,
off
on
    = 1 &,
clear = 0 %;
/*
                                                           */
/* Fower On and Initialization. This process was not modeled but is
  added to initialize signals and registers.
                                                           */
                                                           ¥/
power_on_initialize :=
      SWITCH = on:
                                  ! Turn Power On
      next;
                                  ! Execute Assignment
                                  ! System Not Ready
      READY = lo:
                                  ! Assert Reset For
      RESET = lo:
                                  ! 100 hiliseconds(Active Low)
      delay(100);
      RESET = hi;
                                  ! Deactivate Reset
                                  ! Execute Fending Assignments
      next;
                                  ! Initialize Address Strobe
      ASN = hi;
      LDSN = hi:
                                  ! Initialize Lower Data Strobe
                                  ! Initialize Upper Data Strobe
      UÚSN = hi;
                                  ! Initialize Data Transfer Acknowledge
      INTACKN = hi;
      RW = hi;
                                  ! Initialize Read/Write(Read On High)
      DBUS = 0xffff;
                                 ! Place Data Bus In High Impedance State
      M[0x100c] = 0xff;
                                   ! Place Memory Locations Following The
      ME0 \times 100d] = 0 \times ff;
                                   ! JMF Instruction In A High State
      HALT = hi;
                                  ! Initialize Halt Flip-Flop(Active
                                  ! Low)
      T = 0;
                                  ! Initialize Clock Cycle Counter
      READY = hi;
                                  ! System Ready
      */
      /*
            Routine Initialization Fer Hamby and Guillory
                                                           1/
                                                           */
      /×
      M[0x2006] = 0x55;
                                  ! Data To Be Moved
      ME0 \times 2007] = 0 \times 55;
      M[0x2004] = 0xaa;
      MLOx2005J = Oxaq;
      D(2) = 0x2008;
                                  ! Place 2008 Into DE23
                                  ! Place Hex 1004 Into A[0]
      A[0] = 0 \times 1004;
      A[1] = 0x2008;
                               ! Store Data At This Address
                                  ! Place Hex 1000 Into Program Counter
      PC = 0 \times 1000;
      next
                                  ! Execute Assignments
```

```
/*
                                                        x/
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                        */
/* to retrieve modeled instructions for simulation and analysis. It
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
                                                        */
/* on page VI-15 of their thesis.
                                                        */
fetch_initial_instruction :=
    PHI1 = hi:
                                   ! Phase 1 Of
                                   ! Clock Cycle 0
    FHI2 = 10;
    RW = hi;
                                   ! Memory Read
                                   ! Disable Address Bus Buffer
    ADENABLE = 10:
    DBENABLE = 10;
                                   ! Disable Data Bus Buffer
    TABUS = PC;
                                   ! Place PC On Internal Address
                                   ! Execute Pending Assignments
    next;
    PHI1 = 10;
                                   ! Phase 2 Of
    PHI2 = hi:
                                   ! Clock Cycle 0
    ADENABLE = hi;
                                   ! Enable Address Bus Buffer
    EXABUF = IABUS;
                                   ! Gate Internal Address Bus
                                   ! Into External Address Buffer
    FCMODE = SRMODE;
                                   ! User Mode
    FCSPACE = 2;
                                   ! Accessing Program
                                   ! Execute Impending Assignments
    next;
    ABUS = EXABUF;
                                   ! Address Placed On Rus(Added)
                                   ! Execute Pending Assignments
    next;
    ! Clock Cycle 1
    T = 1;
    next;
                                   ! Execute Assignment
                                   ! Phase 1 Of
    PHI1 = hi;
                                   ! Clock Cycle 1
    PHI2 = 10;
    ASN = lu;
                                   ! Assert Address Strobe
                                   ! Assert Lower Data Strobe
    LIISN = 10:
    UDSN = lo:
                                   ! Assert Upper Data Strobe
    DBENABLE = hi;
                                   ! Enable Nata Rus
                                   ! Execute Pending Assignments
    next;
                                   ! Phase 2
    PHI1 = lo;
    PHI2 = hi;
                                   ! Of Clock Cycle 1
                                   ! Execute Fending Assignments
    next;
    T = 2:
                                  ! Clock Cycle 2
```

next;

! Execute Assignment

```
FHI1 = hi;
                              ! Phase 1
PHI2 = 10;
                              ! Of Clock Cycle 2
                              ! Wait For Memory To Place
while DTACKN eql hi
                              ! Data On The Bus
    (
                              ! Execute Impending Assignments
    next;
    PHI1 = lo;
                             ! Phase 2
                              ! Of Clock Cycle 2
    PHI2 = hi;
    next;
                              ! Execute Assignments
    ! Clock Cycle 3
    next;
                              ! Execute Assignment
                              ! Phase I
    PHI1 = hi;
                              ! Of Clock Cycle 3
    PHI2 = 10;
    DBUS<15:8> = MEABUSJ;
                             ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                             ! On Ilata Bus And
                              ! Asserts DTACKN(Added)
    DTACKN = 10;
                              ! Execute Pending Assignments
    next;
    T = 2
                              ! Return To Phase 2
                              ! Of Clock Cycle 2
    );
                             ! Execute Impending Assignments
    next;
T = 3:
                              ! Clock Cycle 3
                              ! Execute Assignment
next;
                              ! Phase 2
FHI1 = 10;
                              ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DBUS;
                              ! Instruction On Data Bus
                              ! Is flaced In External Data
                              ! Bus Buffer
next;
                              ! Execute Pending Assignments
! Clock Cycle 4
T = 4;
                              ! Execute Assignment
next;
                              ! Phase 1
PHI1 = hi;
                              ! Of Clock Cycle 4
PHI2 = 10;
PFR = EXDBUF;
                              ! The Contents Of The External
                              ! Data Bus Buffer Are Placed
                              ! In Prefetch Register
                              ! Execute Pending Assignments
next;
PHI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 4
FHII2 = hi;
ASN = hi:
                              ! Deactivate Address Strobe
```

```
LDSN = hi;
                                        ! Deactivate Lower Data Strobe
     UDSN = hi;
                                        ! Deactivate Upper Data Strobe
     IR = PFR;
                                        ! Contents Of Frefetch Register
                                         ! Are Pluced Into Instruction
                                        ! Register
     DITACKN = hi;
                                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
     PC = PC + 4;
                                        ! Increment Program Counter
                                        ! Execute Fending Assignments
     next;
     T = 0
                                        ! Reset Clock Cycle Counter
andi :=
                                        ! AND.W #$DFFF,SR
                                        ! Effect Of Instruction
    SRMODE = lo;
    IR<15:8> = MEFC3;
                                        ! Prefetch Next Instruction
    1R<7:0> = MEPC + 13;
                                        ! Is To Set Status Register
    next;
       PC = PC + 2;
                                           ! Increment Program Counter
    T = 5;
                                        ! Supervisor Bit To User
                                        1 Mode
    next:
    T = 0
                                         ! Requires 6 Clock Cycles
                                        ! MOVE.W -(A1),D4
BIOVE :=
     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                        ! Clock Cycle 0
                                        ! Place Data Rus In High Impedance
     DBUS = Oxffff;
     RW = hi:
                                        ! Memory Read
                                        ! Disable Address Bus Buffer
     ADENABLE = 10;
                                        ! Address Bus High Impedanced
     ABUS = 0xffffff;
     DRENABLE = 10:
                                        ! Disable Data Bus Buffer
     TABUS<31:1> = FC<31:1>;
                                        ! Place PC On Internal Address
                                        ! Bus
                                        ! Execute Pending Assignments
     next:
                                        ! Phase 2 Of
     PHI1 = 10;
     PHI2 = hi;
                                        ! Clack Cycle 0
                                        ! Enable Address Bus Buffer
     ADENABLE = hi;
                                        ! Gate Internal Address Bus
     EXABUF = IABUS<23:1>;
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSFACE = 2:
                                        ! Accessing Program
     ABUS = IABUS<23:1>;
                                        ! Address Placed On Bus
                                        ! Execute Impending Assignments
     next:
     ! Clock Cycle 1
     T = 1:
     next;
                                        ! Execute Assignment
```

```
PHI1 = hi;
                              ! Phase 1 Of
                              ! Clock Cycle 1
PHI2 = 10;
                              ! Assert Address Strobe
ASN = 10;
                              ! Assert Lower Data Strobe
LDSN = 10;
                              ! Assert Upper Data Strobe
UIISN = lo;
                              ! Enable Nata Bus
DBENABLE = hi;
                              ! Execute Pending Assignments
next;
                              ! Phase 2
PHI1 = lo:
                              ! Of Clock Cycle 1
PHI2 = hi:
next;
                              ! Execute Pending Assignments
T = 2;
                             ! Clock Cycle 2
                              ! Execute Assignment
next;
                              ! Phase 1
PHI1 = hi;
                              ! Of Clock Cycle 2
FHI2 = 10;
                              ! Wait For Memory To Place
while DTACKN eql hi
                              ! Itata On The Bus
                              ! Execute Impending Assignments
    next;
                              ! Phase 2
    F'HI1 = 10:
    PHI2 = hi;
                              ! Of Clock Cycle 2
    next;
                              ! Execute Assignments
    ! Clock Cycle 3
    T = 3;
     next;
                              ! Execute Assignment
     PHI1 = hi;
                              ! Phase 1
     PHI2 = 15:
                              ! Of Clock Cycle 3
     DBUS<15:8> = MEABUS3;
                              ! Memory Places Instruction
                              ! On Data Bus And
     DBUS<7:0> = MCABUS + 13;
                              ! Asserts DTACKN(Added)
     INTACKN = 10;
                              ! Execute Pending Assignments
     next;
     ! Return To Phase 2
     T = 2
                               ! Of Clock Cycle 2
     );
                               ! Execute Impending Assignments
     next;
! Clock Cycle 3
T = 3;
                               ! Execute Assignment
next;
FHI1 = 10;
                              ! Phase 2
                              ! Of Clock Cycle 3
PHI2 = hi;
                              ! Instruction On Data Bus
EXDBUF = DBUS;
                              ! Is Placed In External Data
                               ! Bus Buffer
```

```
next:
                               ! Execute Pending Assignments
! Clock Cycle 4
T = 4;
                               ! Execute Assignment
next;
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 4
PFR = EXPRUF;
                               ! Prefetch Register Gets External
                               ! Execute Pending Assignments
next;
                               ! Phase 2
FHI1 = lo:
PHI2 = hi;
                               ! Of Clock Cycle 4
                               ! Neactivate Address Strobe
ASN = hi;
                               ! Deactivate Lower Data Strobe
LUSN = hi;
UIISN = hi;
                               ! Neactivate Upper Nata Strobe
                               ! Are Flaced Into Instruction
                               ! Register
PC = PC + 2;
                               ! Increment Program Counter
IITACKN = hi:
                               ! Neactivate Nata Transfer(Added)
                               ! Acknowledge
next;
! Clock Cycle 5
T = 5;
next;
                               ! Execute Previous Assignment
PHI1 = hi;
                               ! Phase 1 Of
FHI2 = 10;
                               ! Clock Cycle 5
                               ! Decrement A[1]
A[1] = A[1] - 2;
ABUS = 0xffffff;
                               ! Address Bus High Impedanced
DBUS = 0xffff;
                               ! Data Bus High Impedanced
                               ! Execute Pending Assignments
next;
FHI1 = lo;
                               ! Phase 2 Of
                               ! Clock Cycle 5
PHI2 = h1:
next;
                               ! Into External Address Buffer
T = 6;
                               ! Clock Cycle 6
                               ! Execute Assignment
next:
PHI1 = hi;
                               ! Phase 1 Of
                               ! Clock Cycle 6
PHI2 = 10;
                               ! Execute Pending Assignments
next;
PHI1 = 10
                               ! Phase 2
                               ! Of Clock Cycle 6
PHI2 = hi;
next:
                               ! Execute Pending Assignments
! Clock Cycle 7
7 = 7;
```

```
! Execute Previous Assignment
next;
                                   ! Phase 1 Of
PHI1 = hi;
                                   ! Clock Cycle 7
PHI2 = 10;
                                   ! Memory Read
RW = hi:
ADENABLE = 10;
                                   ! Disable Address Bus Buffer
                                   ! Disable Data Bus Buffer
DIBENABLE = 10;
                                   ! Place A[1] On Internal Address
IABUS = AE13;
                                   ! Rus
                                   ! Execute Fending Assignments
next;
                                   ! Phase 2 Of
PHI1 = lo;
                                   ! Clock Cycle 7
FHI2 = hi;
                                   ! Enable Address Bus Buffer
ABENABLE = hi;
FCMODE = SRMODE;
                                  ! User Mode
                                  ! Accessing Data
FCSFACE = 1;
EXABUF = IABUS<23:1>;
                                  ! Gate Internal Address Bus
ABUS = IABUS<23:1>;
                                  ! Place Address On Rus
                                   ! Initialize Status Register
SRCARRY = 0;
SROVER = 0;
                                   ! Condition Bits
SRZERO = 0;
SRNEG = U;
                                   ! Into External Address Buffer
next;
T = 8:
                                  ! Clock Cycle 8
next:
                                   ! Execute Assignment
PHI1 = hi;
                                   ! Phase 1 Of
                                   ! Clock Cycle 8
PHI2 = 10;
Ulian = lo:
                                   ! Activate Upper And
LDSN = 10;
                                   ! Lower Data Strobes
                                   ! Assert Address Strobe
ASN = lo;
DBENABLE = hi;
                                   ! Enable Data Bus
                                   ! Execute Pending Assignments
next;
PHII = lo;
                                   ! Phase 2
                                   ! Of Clock Cycle 8
F'HI2 = hi; •
                                   ! Execute Pending Assignments
T = 9;
                                   ! Clock Cycle 9
                                   ! Execute Assignment
next;
                                   ! Phase 1 Of
PHI1 = hi;
                                   ! Clock Cycle 9
PHI2 = lo;
while DTACKN eql hi
                                   ! Wait For Memory To Flace
                                   ! Nata On The Bus
     next;
                                   ! Execute Impending Assignments
     PHI1 = lo;
                                   ! Phase 2
     FHI2 = hi;
                                   ! Of Clock Cycle 9
     next;
                                   ! Execute Assignments
```

The state of the s

```
T = 10:
                                ! Clack Cycle 10
     next;
                                ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1 '
                                ! Of Clock Cycle 10
    PHI2 = 16:
                                ! Nemory Flaces Instruction
    DBUS(15:8> = MCABUS];
     DBUS<7:0> = MCABUS + 13;
                               ! On Data Bus And
    DTACKN = 10;
                                ! Asserts DTACKN(Added)
    next;
                                ! Execute Pending Assignments
     T = 9
                                ! Return To Phase 2
                                ! Of Clock Cycle 9
     );
                                ! Execute impending Assignments
     next;
T = 10:
                                ! Clock Cycle 10
                                ! Execute Assignment
next;
FHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 10
EXDRUF = DRUS;
                               ! Instruction On Nata Bus
                               ! Is Placed In External Data
                               ! Bus Huffer
                                ! Execute Pending Assignments
next;
T = 11;
                                ! Clock Cycle 11
next.;
                                ! Execute Assignment
FHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 11
PHI2 = 10;
IDBUS = EXDBUF:
                               ! Set Condition Code Bits
if EXDRUF eq1 0
  SRZERO = hi;
                                ! As Appropriate
if EXDBUF<15>
  SKNEG = hi:
                                ! Execute fending Assignments
next;
FHI1 = 10;
                                ! Phase 2
                                ! Of Clock Cycle 11
PHI2 = hi;
if IR eal 0x3821
                                ! Place Value In Either
                                ! DC43 Or DC33
  D[4] = IDBUS
                                ! Depending On Instruction
else
  D(3) = IDBUS;
ASN = hi;
                                ! Neactivate Address Strobe
                                ! Deactivate Lower Data Strobe
LUSN = hi;
                                ! Deactivate Upper Data Strobe
UDSN = hi;
ITACKN = hi;
                                ! Deactivate Data Transfer(Added)
                                ! Acknowledge
```

```
IR = PFR;
                                         ! Execute Pending Assignments
     next:
     T = 0
                                        ! MOVE.L D2.A1
movel :=
     ! Phase 1 Of
     PHI1 = hi;
                                         ! Clock Cycle 0
     FHI2 = 10;
                                         ! Disable Address Bus
     ADENABLE = 10;
     DIBENABLE = 10;
                                         ! Disable Data Bus
     DBUS = Oxffff;
                                         ! Place Data Rus In High Impedance
     RW = hi;
                                         ! Memory Read
     IARUS = PC;
                                         ! Place PC On Internal Address
                                         ! Bus
     IDBUS = DC23;
                                      ! Flace Data From D[2] Onto
                                         ! Internal Data Rus
                                         ! Execute Pending Assignments
     next;
     PHI1 = 10;
                                         ! Phase 2 Of
     PHI2 = hi;
                                         ! Clock Cycle O
     ADENABLE = hi;
                                         ! Enable Address Bus Buffer
     EXABUF = TABUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
                                         ! User Mode
     FCMODE = SRMODE;
     FCSFACE = 2;
                                         ! Accessing Program
                                         ! Clear Status Register Carry Bit
     SRCARRY = lo;
                                         ! Clear Status Register Overflow Bit
     SROVER = lo;
     SRZERO = 16;
                                         ! Clear Status Register Zero Bit
     SRNEG = 10;
                                         ! Clear Status Register Negative Bit
     A[1] = IDBUS;
                                      ! Place Data From Internal Data Bus
                                         ! Into A[1]
     next;
                                         ! Execute Impending Assignments
     ABUS = EXABUF;
                                         ! Address Floced On Bus(Added)
                                         ! Execute Pending Assignments
     next;
     T = 1;
                                         ! Clock Cycle 1
                                         ! Execute Assignment
     next;
     PHI1 = hi;
                                         ! Phase 1 Of
     f'HI2 = 10;
                                         ! Clock Cycle 1
                                         ! Assert Address Strobe
     ASN = lo;
     LISN = 10:
                                         ! Assert Lower Data Strobe
     UDSN = lo;
                                         ! Assert Upper Data Strobe
     DBENABLE = hi;
                                         ! Enable Data Bus
     if AC13 eq1 0
                                      ! Set Status Register Zero Bit
                                        ! If Moved Data Is Zero
        SRZERO = hi;
                                         ! Execute Pending Assignments
     next;
```

```
! Phase 2
FH11 = lo;
PH12 = hi:
                            ! Of Clock Cycle 1
if A[1](31)
                            ! Set Status Register Negative
  SRNEG = h1;
                            ! Bit If Moved Data Is Negative
                            ! Execute Pending Assignments
next;
: Clock Cycle 2
                            ! Execute Assignment
next;
                            ! I'huse 1
FHI1 = h_1;
                            1 Of Clock Cycle 2
PHI2 = 16;
                            ! Wait For Memory To Place
while DTACKN eql hi
    (
                            ! Data On The Bus
                            ! Execute Impending Assignments
    next;
    PHI1 = 10;
                            ! Phase 2
                            ! Of Clock Cycle 2
    PHI2 = hi;
    next;
                            ! Execute Assignments
    T = 3;
                            ! Clock Cycle 3
    next;
                            ! Execute Assignment
    PHI1 = hi;
                           ! Phase 1
    PHI2 = 10;
                            ! Of Clock Cycle 3
    DBUS<15:8> = MEABUS3;
                            Memory Flaces Instruction
    IRUS<7:00 = MEARUS + 1];
                            📑 On Data Rus And
                            ! Asserts DTACKN(Added)
    INTACKN = 10;
    next:
                            ! Execute Pending Assignments
    ! Return To Phase 2
    T = 2
                            ! Of Clock Cycle 2
    );
                            ! Execute impending Assignments
    next;
T = 3;
                            ! Clock Cycle 3
                            ! Execute Assignment
next;
                            ! Phase 2
PHI1 = 10;
                            ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DRUS;
                            ! Instruction On Nata Rus
                            ! Is flaced In External Data
                            ! Bus Buffer
                            ! Execute Pending Assignments
! Clock Cycle 4
T = 4:
next;
                             ! Execute Assignment
```

```
PHI1 = bi;
                                         ! Phase 1
     PHI2 = lo;
                                         ! Of Clock Cycle 4
     PFR = EXDBUF;
                                         ! The Contents Of The External
                                         ! Nata Bus Buffer Are Placed
                                         ! In Frefetch Register
                                         ! Execute Pending Assignments
     next;
     PHI1 = 10:
                                         ! Phase 2
                                         ! Of Clock Cycle 4
     PHI2 ≈ hi;
                                         ! Deactivate Address Strobe
     ASN = hi:
     LUSN = hi;
                                         ! Deactivate Lower Data Strobe
     UDSN = hi;
                                         ! Deactivate Upper Data Strobe
                                         ! Contents Of Frefetch Register
     IR = PFR;
                                         ! Are Placed Into Instruction
                                         ! Register
                                         ! Deactivate Data Transfer(Added)
     DTACKN = hi;
                                         ! Acknowledge
     PC = PC + 2;
                                         ! Increment Program Counter
     next:
                                         ! Execute Impending Assignments
     T = 0
                                         ! Reset Clock Cycle Counter
                                         ! JMF (AO)
.;: quit.
     PhI1 = hi;
                                         ! Phase 1 Of
     PHI2 = 10;
                                         ! Clock Cycle 0
     INUS = 0xffff;
                                         ! Place Data Rus In A High Impedance
     KW = hi;
                                         ! Memory Read
     ADENABLE = 16;
                                         ! Disable Address Bus Ruffer
     IIBENABLE = 10;
                                         ! Disable Data Bus Buffer
     IABUS = PC;
                                         ! Place PC On Internal Address
                                         ! Bus
                                         ! Execute Pending Assignments
     next;
     FHI1 = 10;
                                         ! Phase 2 Of
     PHI2 = h1;
                                         ! Clock Cycle O
                                         ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
     FCMODE = SRMODE;
                                         ! User Mode
     FCSFACE = 2;
                                         ! Accessing Program
                                         ! Execute Pending Assignments
     next;
     ABUS = EXABUF;
                                         ! Address Flaced On Bus(Added)
     next;
                                         ! Execute Pending Assignments
     T = 1;
                                         ! Clock Cycle 1
     next:
                                         ! Execute Assignment
     FHI1 = hi;
                                         ! Phase 1 Of
```

```
PHI2 = 10;
                                ! Clock Cycle 1
ASN = lo;
                                ! Assert Address Strobe
LISN = lo;
                                ! Assert Lower Data Strobe
UIISN = lo;
                                ! Assert Upper Data Strobe
1ABUS = ACO3;
                                ! Move Jump Address From A[0]
                              , ! To Internal Address Buffer
DRENABLE = hi:
                                ! Enable Data Bus
next:
                                ! Execute Fending Assignments
PHI1 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 1
PC = IABUS;
                                ! Place Jump Address Into Program
next;
人意或就才求此次求求求求求未未未未未未未未未未未未未未未未未未失失之。
                                ! Clock Cycle 2
next;
                                ! Execute Assignment
                                ! Phase 1
PHI1 = hi;
PHI2 = lo;
                                ! Of Clock Cycle 2
while DTACKN eql hi
                                ! Wait For Memory To Place
                                ! Data On The Rus
     next:
                                ! Execute Impending Assignments
                                ! Phase 2
     PHI1 = 10;
                                ! Of Clock Cycle 2
     PHI2 = hi;
     next;
                                ! Execute Assignments
     T = 3;
                                ! Clock Cycle 3
     next;
                                ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
     PHI2 = 10;
                                ! Of Clock Cycle 3
     DBUS<15:8> = MEABUS3;
                                ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                                ! On Data Bus And
     DITACKN = lo;
                                ! Asserts DTACKN(Added)
                                ! Execute Fending Assignments
     T = 2
                                ! Return To Phase 2
                                ! Of Clock Cycle 2
     );
     next;
                               ! Execute Impending Assignments
T = 3;
                                ! Clock Cycle 3
next;
                                ! Execute Assignment
f'H11 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 3
EXDBUF = DRUS;
                                ! Instruction On Data Bus
```

. .

```
! Is Placed In External Data
                                  ! Bus Buffer
next;
                                  ! Execute Pending Assignments
! Clock Cycle 4
next;
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1
PHI2 = 10;
                                  ! Of Clock Cycle 4
next;
FFR = EXDBUF;
                                  ! The Contents Of The External
                                  ! Nata Bus Buffer Are Placed
                                  ! In Prefetch Register
next;
                                  ! Execute Pending Assignments
                                  ! fhase 2
FHI1 = 10;
                                  ! Of Clock Cycle 4
PHI2 = hi:
ASN = hi;
                                  ! Deactivate Address Strobe
                                  ! Deactivate Lower Data Strobe
LUSN = hi;
                                  ! Neactivate Upper Nata Strobe
UIISN = ha;
                                  ! Deactivate Data Transfer
DYACKN = hi;
                                  ! Acknowledge(Added)
T = 5;
                                  ! Clack Cycle 5
                                  ! Execute Previous Assignment
next;
PHI1 = h1;
                                  ! Phase 1 Of
PHI2 = 16:
                                  ! Clock Cycle 5
RW = hi;
                                  ! hemory Read
                                  ! Disable Address Bus Buffer
ADENABLE = 10;
DEBENABLE = 10;
                                  ! Disable Data Bus Buffer
1ABUS = PC;
                                  ! Place PC On Internal Address
                                  ! Bus
                                  ! Execute Fending Assignments
next;
PHI1 = lo;
                                  ! Phase 2 Of
                                  ! Clock Cycle 5
f'HI2 = hi;
                                  ! Enable Address Bus Buffer
AUENABLE = hi;
                                  ! User Mode
FCMODE = SRMODE;
FOSTACE = 2;
                                  ! Accessing Program
LXABUF = IABUS;
                                  ! Gate Internal Address Bus
next;
                                  ! Into External Address Buffer
ABUS = EXABUF;
                                  ! Address Flaced On Rus(Added)
next:
                                  ! Execute Pending Assignments
T = 6;
                                  ! Clock Cycle 6
next;
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1 Of
PHI2 = 10;
                                  ! Clock Cycle 6
```

```
ASN = 10:
                              ! Assert Address Strobe
LISN = lo;
                              ! Assert Lower Data Strobe
                              ! Assert Upper Data Strobe
UliSN = lo;
                              ! Enable Data Rus
DIRENABLE = hi;
next;
                              ! Execute Pending Assignments
FHI1 = 15:
                              ! Phase 2
PHI2 = hi:
                              ! Of Clock Cycle 6
next:
                              ! Execute Pending Assignments
T = 7;
                              ! Clock Cycle 7
next;
                              ! Execute Assignment
FHI1 = hi;
                              ! Phase 1
                              ! Of Clock Cycle 7
PH12 = 16;
while DTACKN eql hi
                              ! Wait For Memory To Fluce
                             ! Data On The Bus
                             ! Execute Impending Assignments
    next;
    PHI1 = lo:
                              ! Phase 2
                              ! Of Clock Cycle 7
    FHI2 = hi;
                             ! Execute Assignments
    next;
     ! Clock Cycle 8
    7 = 8;
    next;
                              ! Execute Assignment
    PH11 = h1;
                              ! Phase 1
    FHI2 = io;
                              ! Of Clock Cycle 8
    'DBUS(15:8) = MCABUS];
                              ! hemory Flaces Instruction
                              ! On Data Bus And
    I(BUS<7:0> = MEABUS + 13;
    DTACKN = lo;
                              ! Asserts DTACKN(Added)
    next;
                              ! Execute Pending Assignments
     ! Return To Phase 2
    T = 7
                              ! Of Clock Cycle 7
     );
                           ! Execute Impending Assignments
    next:
T = 8:
                              ! Clock Cycle 8
                              ! Execute Assignment
next;
PHI1 = lo;
                              ! Phase 2
PH12 = hi;
                              ! Of Clock Cycle 8
EXDBUF = DBUS;
                              ! Instruction On Data Bus
                              ! Is Placed In External Data
                              ! Rus Buffer
                              ! Execute Pending Assignments
next;
```

```
T = 9;
                                             ! Clock Cycle 9
      next;
                                              ! Execute Assignment
      PHI1 = hi;
                                             ! Phase 1
                                             ! Of Clock Cycle 9
      FHI2 = 10;
      PFR = EXDBUF:
                                             ! The Contents Of The External
                                             ! Note Bus Buffer Are Flaced
                                             ! In Frefetch Register
                                             ! Execute Pending Assignments
      next;
                                             ! Phase 2
      PHI1 = 10;
      PHI2 = hi:
                                             ! Of Clock Cycle 9
      ASN = hi;
                                             ! Deactivate Address Strobe
                                             ! Deactivate Lower Data Strobe
      LDSN = hi;
                                             ! Deactivate Upper Data Strobe
      UDSN = hi;
      F'C = PC + 2;
                                             ! Increment Program Counter.
      IR = PFR;
                                             ! Place Contents Of Prefetch
                                              ! Register Into Instruction
                                             ! Register
      DTACKN = hi;
                                             ! Deactivate Data Transfer
                                             ! Acknowledge(Added)
      next;
                                             ! Execute Pending Assignments
                                             ! Reset Clock Cycle Counter
      T = 0
decode_execute_prefetch !=
                        case IR
                                                    ! MOVE.W -(A1),D4 [D3]
                             0x3821,0x3621: move
                                                    ! AND.W #$DFFF.SR
                                     0x027c: andi
                                     047320: jmp
                                                    ! JMP (AQ)
                                     0x2242: move1 ! MOVE.L D2,A1
                         esac
main :=
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
           decode_execute_prefetch
     )
```

```
/*
                                                */
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W 04(A1), D1 INSTRUCTION
                                                */
/*
                                                1/
/*
                                                */
/#
                                                1/
              Structure Declarations
1*
                                                */
/*
                                                */
/*
                                                */
           M68000 Frogramming Registers
/*
                                                ¥/
DE0:73<31:0>,
                   ! 8 Data Registers
A[0:63<31:0>,
                   ! 7 Address Registers
UA7<31:0>,
                   ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Pointer
PC<31:0>,
                   ! Program Counter
                   ! Status Register
SR<15:0>,
/*
                                                */
/*
                                                */
           Temporary Internal Registers
                                                x/
/*
PFR<15:0>,
                   ! Frefetch Register
IR<15:0>,
                   ! Instruction Register
FC<2:0>,
                   ! Function Code Register
EXDBUF<15:0>,
                   ! External Data Bus Buffer Register
EXABUF<23:1>,
                   ! External Address Bus Buffer Register(changed)
                   ! ALU Buffer 1
ALUBUF1<31:0>,
ALURUF2<31:0>.
                   ! ALU Buffer 2
DTEMP<15:0>.
                   ! Temporary Data Storage
DISREG<31:0>.
                   ! Temporary Displacement Storage
SRTEMP<15:0>,
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPADR<31:0>,
                   ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>.
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>,
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:0>.
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
T<7:0>,
                        ! Clock Cycle Counter
                      ! Reset Flip-Flop
RESET,
HnLT,
                     ! Hult Flip-Flop
Ŕ₩.
                     ! Read/Write Flip-Flop
ADENABLE,
                     ! Address Bus Buffer Enable
LIBENABLE,
                     ! Nata Bus Buffer Enable
ASN.
                     ! Address Strobe Flip-Flop
LUSN.
                     ! Lower Nata Strobe Flip-Flop
ULISN,
                      ! Upper Data Strobe Flip-Flop
LITACKN,
                     ! Nata Transfer Acknowledge Flip-Flop
COUT,
                      ! Carry Flip-Flop
EXCEPT.
                      ! Exception Processing Flip-Flop
READY,
                      ! Ready Flip-Flop
/*
                                                              */
/*
       Model transformation modifications:
                                                              */
/*
                                                              */
/*
           1) CDL decoder structure nonexistent in ISP' and un-
                                                              */
       necessary for model. Eliminated.
/*
                                                              */
/*
           2) Multi-phase clock structure nonexistent in ISP'.
                                                              */
/*
       Operations on registers will provide its equivalent.
                                                              */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
/*
       register will provide its equivalent.
                                                              */
/*
           4) The declared bus structures are modeled with registers */
       without loss of model accurracy. This done to maintain model
1*
                                                              */
       equivalency and simplicity.
                                                              */
/*
/*
           5) The memory word length was reduced from 16 to 8 bit
                                                              */
/*
       words to coincide with the ECR's 32-Kbyte memory, to agree with*/
/*
       their PC incrementation, and to enable the use of existing
                                                              */
/*
       MC68000 assembler and linker/loader models. The memory was
                                                              */
       also reduced from 8 Mwords to 32 Kbytes.
                                                              */
/*
/x
                                                              */
1ABUS<31:0>,
                        ! Internal Address_Bus
IDRUS<31:0>,
                        ! Internal Data Bus
twait<4:0>,
                        ! Wait State Counter
SWITCH,
                     ! Power Switch
FHI1.
                      ! Phase 1 Of Two-Phase Clack
PHI2;
                      ! Phase 2 Of Two-Phase Clock
port
/*
                                                              */
/*
              External Address and Data Rus
                                                              */
/*
                                                              */
· ! External Data Bus
DBUS<15:0>.
```

(2)

```
ARUS <23:1>;
                     ! External Address Bus(changed)
format
/*
/*
                                                         */
                Register Subfields
                                                         1/
/*
! Program Counter Address Field
PCADUR
         = PC(2310),
SRTRACE
         = SR<15>.
                      ! Trace Bit
SAMODE
         = SR<13>,
                      ! Mode Selection Bit
SRCARRY
         = SR<0>,
                      ! Carry Rit
SKOVER
         = SR<1>,
                      ! Overflow Bit
SRZERO
                      ! Zero Bit
         = SR<2>,
SKNEG
         = SR<3>,
                      ! Negative Bit
                      ! Extend Bit
SREX
         = SR<4>,
                      ! Interrupt Mask
SRMASK
         = SR<10:8>.
         = FC<1:0>, .
FICSFIACE
                      ! Memory Access Address Space
FCMODE
         = FC<2>,
                      ! User/Supervisor Mode Bit
P'CLOW
         = PC<15:0>,
                      ! PC Low Word
PCHI
         = PC<31:16>,
                      ! PC High Word
         = DE03<15:0>,
                      ! II[0] Low Word
DOLWORD
DILWORD
         = DE13(15:0),
                      ! DE13 Low Word
D2LWORD
         = DE23<15:0>,
                      ! II[2] Low Word
                      ! D[3] Low Word
DISLWORD
         = D[3]<15:0>,
I:4LWORI
         = D[4]<15:0>.
                      ! IIE43 Low Word
#5LWORD
         = DC53<15:0>,
                      ! DES] Low Word
                      ! BE63 Low Word
II6LWORD
         = D[63<15:0>.
                      ! D[7] Low Word
DZLWOKD
         = D[7]<15:0>,
DISREGHWORD = DISREG(31:16),! DISREG High Word
DISREGLWORD = DISREG(15:0), ! DISREG Low Word
HANADRLOW
         = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI
         = TEMPADR<31:16>;! TEMPADR High Word
memory
*/
/x
/*
                                                         */
                16K 16-Bit Word Internal Memory
                                                         */
/*
ME0:327673<7:0>;
macro
/*
                                                         */
/*
               Logic Level Macros
                                                         */
```

```
/*
= 0 %,
lo
    = 1 %,
hi
off
    = 0 1,
Oυ
    = 1 %,
clear = 0 &;
*/
/*
/* Power On and Initialization. This process was not modeled but is
                                                          */
/* added to initialize signals and registers.
                                                          1/
/×
                                                          */
power_on_initialize :=
                                 ! Turn Power On
      SWITCH = on;
      next;
                                 ! Execute Assignment
                                 ! System Not Ready
      REALIY = 10;
                                 ! Assert Reset For
      KESET = 10;
      delay(100);
                                 ! 100 Miliseconds(Active Low)
      RESET = hi;
                                 ! Deactivate Reset
      next;
                                 ! Execute Pending Assignments
      ASN = hi;
                                 ! Initialize Address Strobe
      LDSN = hi:
                                 ! Initialize Lower Data Strobe
                                 ! Initialize Upper Data Strobe
      UDSN = hi;
                                 ! Initialize Data Transfer Acknowledge
      DTACKN = hi;
                                 ! Initialize Read/Write(Read On High)
      RW = hi:
                                ! Place Data Bus In High Impedance State
      LIBUS = 0xffff;
                                  ! Place Memory Locations Following The
      MEOx100eJ = Oxff;
      ME0x100f3 = 0xff;
                                   ! JMP Instruction In A High State
                                 ! Initialize Halt Flip-Flop(Active
      HALT = hi:
                                 ! Low)
                                 ! Initialize Clock Cycle Counter
      T = 0;
      REALLY = hi:
                                 ! System Ready
      /*
                                                          */
      /*
            Routine Initialization Per Hamby and Guillory
                                                          */
      /*
                                                          */
      ME0x20043 = 0x55;
                                 ! Value To Be Moved
      ME0x2005] = 0x55;
                                 ! Value To Be Moved
      ME0\times20083 = 0\times44
      ME0\times20093 = 0\times00;
                                 ! Place Hex 1004 Into ACOJ
      A[0] = 0x1004;
      A[1] = 0 \times 2000;
                              ! Store Data At This Address
      PC = 0x1000;
                                 ! Place Hex 1000 Into Frogram Counter
                                 ! Execute Assignments
      next
      )
```

```
/¥
                                                           1/
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
/* to retrieve modeled instructions for simulation and analysis. It
                                                           */
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/# on page VI-15 of their thesis.
                                                           11/
                                                           */
/*
fetch_initial_instruction :=
     PHI1 = hi;
                                     ! Phase 1 Of
                                     ! Clock Cycle 0
    fHI2 = 10:
                                    ! Memory Read
    RW = hi;
                                    ! Disable Address Bus Buffer
    ADENABLE = 10;
                                    ! Disable Data Bus Buffer
    DRENABLE = 10;
                                    ! Place PC On Internal Address
    IABUS = PC;
    next;
                                     ! Execute Pending Assignments
    PHI1 = lo:
                                    ! Phase 2 Of
    PHI2 = hi:
                                    ! Clock Cycle 0
    ADENABLE = hi:
                                    ! Enable Address Bus Buffer
    EXABUF = IABUS;
                                    ! Gate Internal Address Bus
                                    ! Into External Address Buffer
     FCMODE = SRMODE;
                                     ! User Mode
    FCSPACE = 2;
                                    ! Accessing Program
                                    ! Execute Impending Assignments
    next:
    ABUS = EXABUF:
                                    ! Address Placed On Rus(Added)
                                     ! Execute Pending Assignments
     next;
     T = 1;
                                    ! Clock Cycle 1
     next;
                                    ! Execute Assignment
                                    ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                     ! Clock Cycle 1
     ASN = 10;
                                    ! Assert Address Strobe
                                    ! Assert Lower Data Strobe
    LISN = 10;
                                    ! Assert Upper Data Strobe
     UDSN = lo:
                                    ! Enable Data Bus
     DBENABLE = hi;
                                     ! Execute Pending Assignments
     next;
    f'HI1 = 10;
                                    ! Phase 2
                                     ! Of Clock Cycle 1
     PHI2 = hi;
                                     ! Execute Pending Assignments
    next;
     T = 2:
                                     ! Clock Cycle 2
```

next;

! Execute Assignment

```
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
PH12 = 10;
                               ! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
    next;
                               ! Execute Impending Assignments
    PHI1 = lo;
                               ! Phase 2
    PH12 = hi;
                               ! Of Clock Cycle 2
    next;
                               ! Execute Assignments
    ! Clock Cycle 3
    next;
                               ! Execute Assignment
    PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 3
    FHI2 = 10;
    DBUS<15:8> = MCABUSJ;
                              ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                               ! On liata Bus And
                               ! Asserts DTACKN(Added)
    DIACKN = 10;
                               ! Execute Pending Assignments
    next;
     ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
                              ! Execute Impending Assignments
     next;
T = 3:
                               ! Clock Cycle 3
                               ! Execute Assignment
next;
PHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DRUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
next:
                               ! Execute Pending Assignments
! Clock Cycle 4
T = 4;
                               ! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi;
                               ! Of Clock Cycle 4
PHI2 = 10;
                               ! The Contents Of The External
PFR = EXDBUF:
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
                               ! Execute Pending Assignments
next;
                               ! Phase 2
PHI1 = lo;
                               ! Of Clock Cycle 4
PHI2 = ha;
ASN = hi;
                               ! Deactivate Address Strobe
                               ! Deactivate Lower Data Strobe
LDSN = hi;
```

```
UDSN = hi;
                                        ! Deactivate Upper Data Strobe
                                        ! Contents Of Prefetch Register
     IR = PFR;
                                        ! Are Placed Into Instruction
                                        ! Register
     DTACKN = hi;
                                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
     PC = PC + 4;
                                        ! Increment Program Counter
                                        ! Execute Pending Assignments
     next;
                                        ! Reset Clock Cycle Counter
     T = 0
                                        ! AND.W #$DFFF,SR
andi :=
    SRMODE = 10:
                                        ! Effect Of Instruction
                                        ! Prefetch Next Instruction
    IR<15:8> = MCPC3;
    1R<7:0> = MEPC + 13;
    next;
                                        ! Is To Set Status Register
       PC = PC + 2;
                                           ! Increment Frogram Counter
    T = 5:
                                        ! Supervisor Rit To User
    next;
                                        ! Mode
                                        ! Requires 6 Clock Cycles
    T = 0
    ١
                                        ! MOVE.W 4(A1).D1 [8(A1).D2]
move :=
     ! Phase 1 Of
     PHII = hi;
     PHI2 = 10;
                                         ! Clock Cycle 0
     DRUS = 0xffff;
                                        ! Place Data Bus In High Impedance
                                        ! Memory Read
     RW = hi;
     ADENABLE = 10;
                                        ! Disable Address Bus Buffer
     DBENABLE = 10;
                                        ! Disable Data Rus Ruffer
     IABUS = PC;
                                        ! Place PC On Internal Address
                                        ! Execute Pending Assignments
     next;
                                        ! Phase 2 Of
     FHI1 = 10;
                                        ! Clock Cycle 0
     PHI2 = hi;
                                        ! Enable Address Bus Buffer
     ADENABLE = ba:
     EXABUF = IABUS:
                                        ! Gate Internal Address Rus
                                        ! Into External Address Buffer
                                        ! User Mode
     FCMODE = SRMODE:
     FCSPACE = 2;
                                        ! Accessing Program
     ABUS = IABUS;
                                         ! Address Placed On Rus
                                        ! Execute Impending Assignments
     next;
     ! Clock Cycle 1
     T = 1;
                                         ! Execute Assignment
     next;
                                         ! Phase 1 Of
     PHI1 = hi; .
```

```
! Clock Cycle 1
PHI2 = 10;
                               ! Assert Address Strobe
ASN = 10;
LDSN = lo;
                               ! Assert Lower Data Strobe
UIISN = lo;
                               ! Assert Upper Data Strobe
DBENABLE = hi;
                               ! Enable Data Bus
                               ! Execute Pending Assignments
next;
                               ! Phase 2
PHI1 = 10;
                               ! Of Clock Cycle 1
PHI2 = hi;
                               ! Execute Pending Assignments
next;
T = 2:
                               ! Clock Cycle 2
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
FHI2 = 10;
                               ! Wait For Memory To Place
while DTACKN eql hi
                               ! Nata On The Bus
     next;
                               ! Execute Impending Assignments
                               ! Phase 2
     FHI1 = lo;
                               ! Of Clock Cycle 2
     PHI2 = hi;
     next;
                               ! Execute Assignments
     T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
     next;
                               ! Phase 1
     PHI1 = hi;
                               ! Of Clock Cycle 3
     PHI2 = lo;
     DBUS<15:8> = MCABUS3;
                               ! Memory Places Instruction
     DBUS<7:0> = MCABUS + 13;
                               ! On Diata Bus And
     DTACKN = lo;
                               ! Asserts DTACKN(Added)
     next;
                               ! Execute Pending Assignments
     ! Return To Phase 2
     T = 2
                               ! Of Clock Cycle 2
     );
                               ! Execute Impending Assignments
     next;
T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
next;
                               ! Phase 2
FHI1 = 10;
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                               ! Instruction On Data Rus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
```

```
! Clock Cycle 4
T = 4;
                                  ! Execute Assignment
next;
PHI1 = hi;
                                 ! Phase 1
                                 ! Of Clock Cycle 4
PHI2 = 16;
DISREG = EXDBUF sxt 32;
                                 ! Store Displacement
next:
                                 ! Execute Pending Assignments
FHI1 = 10;
                                 ! Phase 2
                                 ! Of Clock Cycle 4
PHI2 = hi;
                                 ! Deactivate Address Strobe
ASN = hi;
                                 ! Deactivate Lower Data Strobe
LIBN = hi;
                                 ! Deactivate Upper Data Strobe
UDSN = hi;
                                  ! Are Flaced Into Instruction
                                 ! Register
PC = PC + 2;
                                 ! Increment Program Counter
DISREG = DISREG + A[1];
                                ! Add Address Register To Displacement
DTACKN = hi:
                                 ! Deactivate Data Transfer(Added)
                                 ! Acknowledge
next:
T = 5;
                                  ! Clock Cycle 5
next:
                                  ! Execute Previous Assignment
                                 ! Phase 1 Of
PHI1 = hi:
                                 ! Clock Cycle 5
PHI2 = lo;
RW = hi:
                                 ! Memory Read
ADENABLE = lo;
                                 ! Disable Address Bus Buffer
DBUS = 0xffff;
                                 ! Data Bus Returned To High
                                 ! Impedance State
                                 ! Disable Data Bus Buffer
DEENABLE = 10;
                                 ! Place PC On Internal Address
IABUS = PC;
                                 ! Execute Pending Assignments
next;
PHI1 = 10;
                                 ! Phase 2 Of
                                 ! Clock Cycle 5
PHI2 = hi:
                                 ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                 ! User Mode
                                 ! Accessing Data
FCSPACE = 2;
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
                                 ! Place Address On Bus
ARUS = IABUS;
next:
                                  ! Into External Address Buffer
T = 6:
                                 ! Clock Cycle 6
                                 ! Execute Assignment
next;
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 6
UDSN = lo;
                                 ! Activate Upper And
```

```
LUSN = 10;
                             ! Lower Data Strobes
                             ! Assert Address Strobe
ASN = lo;
                             ! Enable Data Bus
DBENABLE = h1;
next;
                             ! Execute Fending Assignments
                             ! Phase 2
FHI1 = 1c;
                             ! Of Clock Cycle 6
PHI2 = hi;
                             ! Execute Pending Assignments
next:
T = 7;
                            ! Clock Cycle 7
                             ! Execute Assignment
next;
PHI1 = hi;
                            ! Phase 1 Of
                            ! Clock Cycle 7
FHI2 = 10;
                            ! Wait For Memory To Place
while DTACKN eql hi
                            ! Nata On The Rus
    (
    next:
                            ! Execute Impending Assignments
                            ! Phase 2
    PHI1 = 10;
    PHI2 = hi;
                            ! Of Clock Cycle 7
                             ! Execute Assignments
    next;
    T = 8;
                             ! Clock Cycle 8
                             ! Execute Assignment
    next;
                             ! Phase 1
    PHI1 = hi;
                             ! Of Clock Cycle 8
    PHI2 = 10:
    DBUS<15:8> = MEABUS3;
                             ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                            ! On Data Bus And
                             ! Asserts DTACKN(Added)
    INTACKN = 10;
    next;
                             ! Execute Pending Assignments
    T = 7
                             ! Return To Phase 2
                             ! Of Clock Cycle 7
    );
    next;
                             ! Execute Impending Assignments
T = 8:
                             ! Clock Cycle 8
                             ! Execute Assignment
next;
                             ! Phase 2
f'HI1 = lo;
PHI2 = hi;
                            ! Of Clock Cycle 8
EXDRUF = DRUS;
                            ! Instruction On Data Bus
                            ! Is Placed In External Data
                             ! Bus Buffer
                             ! Execute Pending Assignments
next;
```

T = 9;

! Clock Cycle 9

```
! Execute Assignment
next;
                                   ! Phase 1
PHI1 = hi;
FHI2 = 10;
                                   ! Of Clock Cycle 9
PFR = EXDBUF:
                                   ! The Contents Of The External
                                   ! Data Bus Buffer Are Placed
                                   ! In Prefetch Register
                                   ! Execute Pending Assignments
next;
PHI1 = 10;
                                   ! Phase 2
                                   ! Of Clock Cycle 9
PHI2 = hi;
                                   ! Deactivate Address Strobe
ASN = hi;
LISN = hi;
                                   ! Neactivate Lower Nata Strobe
                                   ! Deactivate Upper Data Strobe
UDSN = hi;
FC = FC + 2;
                                   ! Increment PC
INTACKN = hi;
                                   ! Deactivate Data Transfer(Added)
                                   ! Acknowledge
                                   ! Execute Pending Assignments
next;
T = 10:
                                   ! Clock Cycle 10
                                   ! Execute Previous Assignment
next;
                                   ! Phase 1 Of
PHI1 = hi;
                                   ! Clock Cycle 10
PHI2 = 10;
                                   ! Mesory Read
RW = hi;
                                   ! Disable Address Bus Buffer
ADENABLE = 10;
DBUS = 0xffff;
                                   ! Data Bus Returned To High
                                   ! Impedance State
                                   ! Disable Data Bus Buffer
IRENABLE = 10;
TABUS = DISREG;
                                   ! Place DISREG[1] On Internal
                                   ! Address Bus
                                   ! Execute Pending Assignments
next;
                                   ! Phase 2 Of -
PHI1 = lo;
                                   ! Clock Cycle 10
PHI2 = hi:
ADENABLE = hi;
                                   ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                   ! User Mode
FCSPACE = 1;
                                   ! Accessing Data
EXABUF = IABUS;
                                   ! Gate Internal Address Bus
                                   ! Place Address On Bus
ABUS = IABUS;
SRCAKRY = 0;
                                   ! Initialize Status Register
SROVER = 0;
                                   ! Condition Bits
SRZEKO = 0;
SKNEG = 0;
                                   ! Into External Address Buffer
next;
! Clock Cycle 11
T = 11;
next;
                                    ! Execute Assignment
PH11 = hi;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 11
```

```
UDSN = 10:
                              ! Activate Upper And
                              ! Lower Data Strobes
LI(SN = 10;
                              ! Assert Address Strobe
ASN = 10;
DBENABLE = hi;
                              ! Enable Data Bus
                              ! Execute Fending Assignments
next;
                             ! Phase 2
PHI1 = 10;
PHI2 = hi;
                              ! Of Clock Cycle 11
                              ! Execute Pending Assignments
next:
T = 12:
                              ! Clock Cycle 12
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1 Of
FHI2 = 10;
                             ! Clock Cycle 12
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Itata On The Rus
                             ! Execute Impending Assignments
    next;
    PHI1 = lo;
                             ! Phase 2
                             ! Of Clock Cycle 12
    PHI2 = hi;
                              ! Execute Assignments
    next;
    7 = 13;
                              ! Clock Cycle 13
                             ! Execute Assignment
    next;
    PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 13
    PHI2 = 10;
                              ! Memory Places Instruction
    INBUS<15:8> = MEARUSD;
    DBUS(7:0) = MEABUS + 13;
                              ! On Data Bus And
                              ! Asserts DTACKN(Added)
    DITACKN = 10:
    next:
                              ! Execute Pending Assignments
     T = 12
                              ! Return To Phase 2
                              ! Of Clock Cycle 12
     );
    next;
                             ! Execute Impending Assignments
! Clock Cycle 13
T = 13;
                              ! Execute Assignment
next;
                             ! Phase 2
PHI1 = lo;
                              ! Of Clock Cycle 13
PHI2 = hi;
EXUBUF = INUS;
                              ! Instruction On Nata Bus
                              ! Is Placed In External Data
                              ! Bus Buffer
                              ! Execute Pending Assignments
next;
```

```
T = 14;
                                           ! Clock Cycle 14
     next;
                                           ! Execute Assignment
     PHI1 = hi;
                                          ! Phase 1
     PHI2 = 16;
                                          ! Of Clock Cycle 14
     IDBUS = EXDBUF:
     if EXHBUF eq. 0
                                          ! Set Condition Code Bits
        SRZERO = hi;
                                          ! As Appropriate
     if EXDBUF<15>
        SRNEG = hi;
                                          ! Execute Pending Assignments
     next;
     PHI1 = lo;
                                          ! Phase 2
     PHI2 = ni;
                                          ! Of Clock Cycle 14
     if IR eq1 0x3229
                                          ! Place Value in Either
        D[1] = IDBUS
                                          ! D[13 Or D[23
                                          ! Depending On Instruction
     else
        DI20 = IDBUS;
     ASN ≈ hi;
                                          ! Neactivate Address Strobe
                                          ! Deactivate Lower Data Strobe
     LDSN = hi;
     UDSN = hi;
                                          ! Deactivate Upper Data Strobe
                                           ! Deactivate Data Transfer(Added)
     DTACKN = hi;
                                          ! Acknowledge
     IR = PFR;
                                          ! Execute Pending Assignments
     next;
     T = 0
                                          ! JMF (AO)
Jiip :=
     PHI1 = hi;
                                          ! Phase 1 Of
     FHI2 = 10;
                                           ! Clock Cycle 0
     DBUS = 0xffff;
                                          ! Place Data Bus In A High Impedance
                                          ! Memory Read
     RW = hi;
                                          ! Disable Address Bus Buffer
     ADENABLE = 10;
     INENABLE = lo:
                                          ! Disable Data Bus Buffer
     IABUS = PC:
                                          ! Place PC On Internal Address
                                           ! Rus
     next;
                                          ! Execute Pending Assignments
     FHI1 = 10;
                                          ! Phase 2 Of
                                          ! Clock Cycle 0
     PHI2 = hi;
                                          ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IARUS;
                                          ! Gate Internal Address Bus
                                          ! Into External Address Buffer
     FCMODE = SRMODE;
                                          ! User Mode
     FCSFACE = 2;
                                          ! Accessing Program
     next;
                                          ! Execute Pending Assignments
     ARUS = EXABUF;
                                           ! Address Placed On Bus(Added)
```

```
next;
                               ! Execute Pending Assignments
T = 1:
                               ! Clock Cycle 1
next;
                               ! Execute Assignment
                               ! Phase 1 Of
PHI1 = hi:
PHI2 = 1c;
                               ! Clock Cycle 1
ASN = lo:
                               ! Assert Address Strobe
LISN = lo:
                              ! Assert Lower Data Strobe
                               ! Assert Upper Data Strobe
UDSN = lo:
IABUS = A[0];
                               ! Move Jump Address From A[0]
                              ! To Internal Address Buffer
DRENABLE = h1:
                               ! Enable Data Bus
next;
                              ! Execute Pending Assignments
PHI1 = lo;
                              ! Phase 2
                               ! Of Clock Cycle 1
PHI2 = hi;
PC = IABUS;
                              ! Place Jump Address Into Program
                               ! Counter
next;
T = 2;
                               ! Clock Cycle 2
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                              ! Of Clock Cycle 2
FHI2 = 10;
                              ! Wait For Memory To Flace
while DTACKN eql hi
                              ! Data On The Bus
                              ! Execute Impending Assignments
    next;
    PHI1 = lo:
                              ! Phase 2
    PHI2 = hi;
                              ! Of Clock Cycle 2
    next;
                               ! Execute Assignments
    T = 3;
                              ! Clock Cycle 3
    next;
                              ! Execute Assignment
    PHI1 = hi;
                              ! Phase 1
    PHI2 = lo;
                              ! Of Clock Cycle 3
                              ! Memory Places Instruction
    DRUS<15:8> = MEARUS];
                              ! On Data Bus And
    DBUS<7:0> = MCABUS + 1];
    DITACKN = 10;
                              ! Asserts ITACKN(Added)
    next;
                              ! Execute Pending Assignments
    ! Return To Phase 2
    T = 2
                               ! Of Clock Cycle 2
    );
    next:
                               ! Execute Impending Assignments
```

```
T = 3;
                                 ! Clock Cycle 3
next;
                                  ! Execute Assignment
FHI1 = 10;
                                 ! Phase 2
                                 ! Of Clock Cycle 3
PHI2 = hi;
                                 ! Instruction On Data Rus
EXERUF = DRUS;
                                 ! Is Placed In External Data
                                 ! Bus Buffer
                                 ! Execute Fending Assignments
next:
T = 4:
                                 ! Clock Cycle 4
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
                                 ! Of Clock Cycle 4
FHI2 = 10;
next:
FFR = EXDBUF:
                                 ! The Contents Of The External
                                 ! Data Bus Buffer Are Placed
                                 ! In Prefetch Register
                                 ! Execute Pending Assignments
next;
PHI1 = 10;
                                 ! Phase 2
                                 ! Of Clock Cycle 4
FHI2 = hi;
                                 ! Deactivate Address Strobe
ASN = hi;
LDSN = hi;
                                 ! Deactivate Lower Data Strobe
UDSN = hi:
                                 ! Deactivate Upper Data Strobe
DTACKN = hi;
                                  ! Deactivate Data Transfer
                                  ! Acknowledge(Added)
T = 5;
                                  ! Clock Cycle 5
next;
                                  ! Execute Previous Assignment
PHI1 = hi;
                                 ! Phase 1 Of
                                  ! Clock Cycle 5
PHI2 = 10;
                                 ! Memory Read
RW = hi;
                                 ! Disable Address Bus Buffer
ADENABLE = 10;
                                 ! Disable Data Bus Buffer
DBENABLE = 10;
                                 ! Place PC On Internal Address
IABUS = PC;
next:
                                  ! Execute Fending Assignments
                                 ! Phase 2 Of
PHI1 = lo;
                                  ! Clock Cycle 5
PHI2 = hi:
                                 ! Enable Address Bus Buffer
ADENABLE = hi;
                                 ! User Mode
FCMODE = SRMODE:
                                 ! Accessing Program
FCSPACE = 2;
                                 ! Gate Internal Address Bus
EXABUF = IABUS;
                                 ! Into External Address Ruffer
next;
ABUS = EXABUF;
                                 ! Address Placed On Bus(Added)
next;
                                 ! Execute Pending Assignments
```

```
T = 6;
                             ! Clock Cycle 6
next;
                             ! Execute Assignment
                             ! Phase 1 Of
PHI1 = hi;
                             ! Clock Cycle 6
FHI2 = 10;
ASN = lo:
                             ! Assert Address Strobe
LISN = lo;
                            ! Assert Lower Nata Strobe
UDSN = lo:
                            ! Assert Upper Data Strobe
IBENABLE = hi:
                            ! Enable Data Rus
next;
                             ! Execute Pending Assignments
PHI1 = 16:
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 6
next;
                            ! Execute Pending Assignments
T = 7;
                             ! Clock Cycle 7
                             ! Execute Assignment
next;
PHI1 = hi:
                             ! Phase 1
                             ! Of Clock Cycle 7
PHI2 = 10;
while DTACKN eql hi
                            ! Wait For Memory To Place
    (
                            ! Data On The Bus
                            ! Execute Impending Assignments
    next;
    PHI1 = 10:
                            ! Phase 2
    PHI2 = hi;
                            ! Of Clock Cycle 7
                            ! Execute Assignments
    next;
    T = 8;
                            ! Clock Cycle 8
    next;
                             ! Execute Assignment
    PHI1 = hi;
                            ! Phase 1
                            ! Of Clock Cycle 8
    FHI2 = 10
    DBUS<15:8> = MCABUSJ;
                             ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                            ! On Data Bus And
    DTACKN = lo:
                            ! Asserts DTACKN(Added)
                            ! Execute Pending Assignments
    next;
    ! Return To Phase 2
    T = 7
                            ! Of Clock Cycle 7
    );
                            ! Execute Impending Assignments
    next;
T = 8:
                             ! Clock Cycle 8
next;
                            ! Execute Assignment
                            ! Phuse 2
PHI1 = lo;
```

```
PHI2 = hi;
                                          ! Of Clock Cycle 8
     EXDRUF = DRUS;
                                          ! Instruction On Data Bus
                                          ! Is Placed In External Data
                                          ! Bus Buffer
                                          ! Execute Pending Assignments
     next;
      ! Clock Cycle 9
     T = 9;
     next;
                                          ! Execute Assignment
     PHI1 = hi;
                                          ! Phase 1
     FHI2 = 10;
                                          ! Of Clock Cycle 9
     PFR = EXIBUF;
                                          ! The Contents Of The External
                                          ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
                                          ! Execute Fending Assignments
     next;
     PHI1 = lo;
                                          ! Phase 2
                                          ! Of Clock Cycle 9
     PH12 = hi;
     ASN = hi;
                                        · ! Deactivate Address Strobe
     LIISN = hi;
                                          ! Deactivate Lower Data Strobe
                                          ! Deactivate Upper Data Strobe
     UDSN = hi;
                                          ! Increment Program Counter
     FC = PC + 2;
                                          ! Place Contents Of Prefetch
     IR = PFR;
                                          ! Register Into Instruction
                                          ! Register
     DTACKN = hi;
                                          ! Deactivate Data Transfer
                                          ! Acknowledge(Added)
                                          ! Execute Pending Assignments
     next;
     T = 0
                                          ! Reset Clock Cycle Counter
     )
decode_execute_prefetch :=
                       case IR
                            0x3229,0x3429; move ! MDVE.W 4(A1),D1 [8(A1),D2]
                            0x027c; andi ! AND.W #$DFFF,SR
                            047320: Jinp
                                         ! JMP (A0) If IR = Octal Value
                       esac
                       )
main :=
    power_on_initialize;
    fetch_initial_instruction;
    while READY eql hi
          decode_execute_prefetch
```

```
/*
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W 04(A1.D7).D2 INSTRUCTION
                                                */
11
                                                1/
/*
                                                */
/*
              Structure Declarations
                                                */
/*
                                                */
state
/×
                                                */
/*
                                                */
           M68000 Programming Registers
/*
                                                */
DE0:73<31:0>,
                  ! 8 Data Registers
AE0:63<31:0>.
                  ! 7 Address Registers
UA7<31:0>,
                  ! User Stack Pointer
SA7<31:0>.
                  ! System Stack Pointer
PC<31:0>,
                  ! Program Counter
SR<15:0>,
                  ! Status Register
/*
                                                */
/*
                                                */
           Temporary Internal Registers
/*
                                                */
PFR<15:0>,
                  ! Frefetch Register
IR<15:0>.
                  ! Instruction Register
FC<2:0>.
                  ! Function Code Register
EXDBUF<15:0>,
                  ! External Data Bus Buffer Register
EXABUF<23:1>,
                  ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,
                  ! ALU Buffer 1
ALUBUF2<31:0>,
                  ! ALU Buffer 2
DTEMP(15:0).
                  ! Temporary Data Storage
I/ISREG<31:0>,
                    Temporary Displacement Storage
SRTEMP<15:0>.
                  ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                  ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPADR<31:0>,
                   ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>.
                  ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>.
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:0>,
                        ! Temporary Address Storage For
                        ! Exception Handler Koutine
                        ! Clock Cycle Counter
T<7:0>,
                      ! Reset Flip-Flop
KESET,
HALT,
                      ! Halt Flip-Flop
RW,
                      ! Read/Write Flip-Flop
                      ! Address Bus Buffer Enable
ADENABLE.
DBENABLE.
                      ! Data Bus Buffer Enable
ASN,
                      ! Address Strobe Flip-Flop
LIISN.
                      ! Lower Data Strobe Flip-Flop
UDSN.
                      ! Upper Data Strope Flip-Flop
DITACKN,
                      ! Nata Transfer Acknowledge Flip-Flop
COUT,
                      ! Carry Flip-Flop
EXCEPT.
                      ! Exception Processing Flip-Flop
READY,
                      ! Ready Flip-Flop
/*
/*
       Model transformation modifications:
                                                              */
                                                              */
/x
           1) CDL decoder structure nonexistent in ISP' and un-
/*
                                                              */
/x
       necessary for model. Eliminated.
                                                              x /
/ X
           2) Multi-phase clock structure nonexistent in ISP'.
                                                              k/
/*
       Operations on registers will provide its equivalent.
                                                              */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
/*
       register will provide its equivalent.
                                                              */
/*
           4) The declared bus structures are modeled with registers */
       without loss of model accurracy. This done to maintain model
                                                              */
/*
/*
       equivalency and simplicity.
                                                              */
                                                              */
/¥
           5) The memory word length was reduced from 16 to 8 bit
/*
       words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*
       their PC incrementation, and to enable the use of existing
                                                              x/
/*
      MC68000 assembler and linker/loader models. The memory was
                                                              */
/*
       also reduced from 8 Mwords to 32 Kbytes.
                                                              1/
/*
                                                              */
IABUS<31:0>,
                        ! Internal Address Rus
IDBUS<31:0>,
                        ! Internal Data Bus
                        ! Wait State Counter
twait<4:0>,
SWITCH.
                      ! Power Switch
FHI1.
                      ! Phase 1 Of Two-Phase Clock
PHI2:
                      ! Phase 2 Of Two-Phase Clock
port
/ X
                                                              k/
/*
             External Address and Data Rus
                                                              */
/ k
                                                              */
DBUS<15:0>,
                        ! External Data Bus
```

```
ABUS<23:1>;
                      ! External Address Bus(changed)
formut
/*
                                                         x/
14
                keqister Subfields
                                                         */
/*
                                                         */
PCALIDR
         = PC<23:0>.
                      ! Program Counter Address Field
SRTRACE
         = SR<15>,
                      ! Trace Bit
SKMODE
         = SR<13>,
                      ! Mode Selection Rit
SRCARRY
         = SR<0>,
                      ! Carry Bit
SROVER
                      ! Overflow Bit
         = SR<1>,
                      ! Zero Bit
SRZERO
         ≈ SR<2>,
SKNEG
         = SR<3>,
                      ! Negative Bit
SREX
         = SR<4>.
                      ! Extend Bit
SRMASK
         = SR<10:8>,
                      ! Interrupt Mask
FCSPACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCMODE
         = FC(2),
                      ! User/Supervisor Mode Bit
F*CLOW
         = PC<15:0>,
                      ! PC Low Word
PCHI
         = PC<31:16>,
                      ! PC High Word
DOLWORD
         = DE03<15:0>.
                      ! II[0] Low Word
DILWORD
         = DC13<15:0>,
                      ! D[13 Low Word
I:2LWORI)
                      ! I/[2] Low Word
         = DC23<15:0>,
D3LWORD
                      ! DE33 Low Word
         = DC33<15:0>,
D4LWORD
         = D[4]<15:0>,
                      ! II(4) Low Word
                      ! DES] Low Word
DSLWORD
         = DE53<15:0>,
LI6LWORD
         = D[6]<15:0>.
                      ! DE63 Low Word
D7LWORD
         = DE73<15:0>,
                      ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANATIRLOW
         = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPAURHI
         = TEMPADR<31:16>;! TEMPADR High Word
Benory
*/
/*
/*
                 16K 16-Rit Word Internal Memory
                                                         */
/#
                                                         */
ME0:327673<7:0>;
BIGCTO
1/
/*
/#
                                                         */
               Logic Level Macros
```

```
10
    = 0 1,
    = 1 8.
hi
    = 0 %,
off
on
    = 1 2,
clear = 0 %;
*/
/*
                                                        */
/* Fower On and Initialization. This process was not modeled but is
/* added to initialize signals and registers.
                                                        */
/*
                                                        */
power_on_initialize :=
      SWITCH = on;
                                ! Turn Power On
                                ! Execute Assignment ..
      next;
      READY = lo:
                                ! System Not Ready
      RESET = lo:
                                ! Assert Keset For
      delay(100);
                                ! 100 Miliseconds(Active Low)
                                ! Deactivate Reset
      RESET = hi;
                                ! Execute Pending Assignments
      next;
      ASN = hi;
                                ! Initialize Address Strobe
      LDSN = hi:
                                ! Initialize Lower Data Strobe
      USISN = hi;
                                ! Initialize Upper Data Strobe
      INTACKN = hi;
                                ! Initialize Data Transfer Acknowledge
                                ! Initialize Read/Write(Read On High)
      RW = hi;
      IIBUS = 0xffff;
                               ! Place Data Bus In High Impedance State
      ME0x100eJ = 0xff;
                                 ! Place Memory Locations Following The
      ME0 \times 100 f = 0 \times f f;
                                  ! JMP Instruction In A High State
      HALT = hi;
                                ! Initialize Halt Flip-Flop(Active
                                ! Low)
      T = 0;
                                ! Initialize Clock Cycle Counter
      READY = hi;
                                ! System Ready
      /*
                                                        */
      /*
           Routine Initialization Fer Hamby and Guillory
                                                        */
      /*
                                                        */
      I([7] = 0 \times 00000006;
                             ! Place 6 Into DC73
      A[0] = 0x1004;
                                ! Place Hex 1004 Into A[0]
      A[1] = 0 \times 2000;
                             ! Store Data At This Address
      M[0x200a] = 0x55;
                              ! Data To Be Moved
      ME0x200b3 = 0x55;
      F'C = 0x1000;
                                ! Place Hex 1000 Into Program Counter
      next
                                ! Execute Assignments
/*
```

で大きない。

```
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
/* to retrieve modeled instructions for simulation and analysis. It
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/# on page VI-15 of their thesis.
                                                           */
/ ¥
fetch_initial_instruction :=
     PHI1 = hi;
                                     ! Phase 1 Of
    PHI2 = 10;
                                     ! Clock Cycle 0
     RW = hi;
                                     ! Memory Read
     ADENABLE = 10;
                                     ! Disable Address Bus Buffer
     DRENABLE = 10;
                                     ! Disable Data Bus Buffer
     IABUS = PC;
                                     ! Place PC On Internal Address
                                     ! Bus
                                     ! Execute Pending Assignments
    next;
     PHI1 = lo;
                                     ! Phase 2 Of
    PHI2 = hi:
                                     ! Clock Cycle 0
     ADENABLE = hi;
                                     ! Enable Address Bus Buffer
    EXABUF = IABUS;
                                     ! Gate Internal Address Bus
                                    ! Into External Address Buffer
     FCMODE = SRMODE;
                                    ! User Mode
     FCSPACE = 2;
                                    ! Accessing Program
                                     ! Execute Impending Assignments
     next;
     ABUS = EXABUF;
                                     ! Address Placed On Bus(Added)
                                     ! Execute Pending Assignments
     next;
     T = 1;
                                     ! Clock Cycle 1
     next;
                                     ! Execute Assignment
                                     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                     ! Clock Cycle 1
                                     ! Assert Address Strobe
     ASN = 10;
     LUSN = lo;
                                    ! Assert Lower Data Strobe
                                     ! Assert Upper Data Strobe
     UDSN = 10;
     DBENABLE = hi;
                                     ! Enable Data Bus
                                     ! Execute Pending Assignments
     next;
                                     ! Phase 2
     PHI1 = lo;
                                     ! Of Clock Cycle 1
     PHI2 = hi;
     next;
                                     ! Execute Pending Assignments
     T = 2;
                                    ! Clock Cycle 2
     next;
                                     ! Execute Assignment
     PHI1 = hi;
                                     ! Phase 1
```

```
! Of Clock Cycle 2
PHI2 = 10:
                               ! Wait For hemory To Place
while INTACKN eql hi
                               ! Data On The Bus
                               ! Execute Impending Assignments
    next;
     PHI1 = lo;
                               ! Phase 2
    PHI2 = hi;
                               ! Of Clock Cycle 2
                               ! Execute Assignments
     next;
     T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
    next;
    PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 3
    PHI2 = 10;
    DBUS<15:8> = MEABUSJ;
                               ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                               ! On Bata Bus And
                               ! Asserts DTACKN(Added)
    DTACKN = 10:
                               ! Execute Fending Assignments
    next;
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
    next;
                               ! Execute Impending Assignments
7 = 3:
                               ! Clock Cycle 3
                               ! Execute Assignment
next;
                               ! Phase 2
PHI1 = lo;
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                               ! Instruction On Data Rus
                               ! Is Placed In External Data
                               ! Bus Ruffer
                               ! Execute Pending Assignments
next;
T = 4;
                               ! Clock Cycle 4
next;
                                ! Execute Assignment
                               ! Phuse 1
PHI1 = hi;
                               ! Of Clock Cycle 4
PHI2 = 10;
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Ruffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
PHI1 = 10;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 4
                               ! Deactivate Address Strobe
ASN = hi;
                               ! Deactivate Lower Data Strobe
LDSN = hi;
                               ! Deactivate Upper Data Strobe
UDSN = hi;
```

```
IR = PFR:
                                        ! Contents Of Prefetch Register
                                        ! Are Placed Into Instruction
                                        ! Register
     INTACKN = hi;
                                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
     PC = PC + 4;
                                        ! Increment Program Counter
                                        ! Execute Pending Assignments
     next:
     T = 0
                                        ! Reset Clock Cycle Counter
     )
andi :=
                                        ! AND W #$DFFF,SR
    SAMODE = 10;
                                        ! Effect Of Instruction
    IR<15:8> = MEPCJ;
                                        ! Prefetch Next Instruction
    1R<7:0> = MEPC + 13;
    next;
                                        ! Is To Set Status Register
      PC = PC + 2;
                                           ! Increment Program Counter
    7 = 5:
                                        ! Supervisor Bit To User
                                        ! Mode
    next;
    T = 0
                                        ! Requires 6 Clock Cycles
                                        ! MOVE.W 4(A1,D7),D2 CD33
BIOVE :=
     PHI1 = hi;
                                        ! Phase 1 Of
     PHI2 = 10:
                                        ! Clock Cycle 0
     DBUS = 0xffff;
                                        ! Place Data Bus In High Impedance
     RW = hi;
                                        ! Memory Read
     ADENABLE = 10;
                                       ! Disable Address Bus Buffer
     ABUS = 0xffffff;
                                       ! Address Bus High Impedanced
                                       ! Disable Data Bus Buffer
     DBENABLE = 10;
     IABUS<31:1> = PC<31:1>;
                                        ! Place FC On Internal Address
                                        ! Bus
     next;
                                        ! Execute Pending Assignments
                                        ! Phase 2 Of
     FHI1 = 10;
                                       ! Clock Cycle O
     PHI2 = hi:
                                       ! Enable Address Bus Buffer
     ADENABLE = hi;
                                        ! Gate Internal Address Bus
     EXABUF = IABUS<23:1>;
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSPACE = 2;
                                        ! Accessing Program
     ABUS = IABUS<23:1>;
                                        ! Address Placed On Bus
                                        ! Execute Impending Assignments
     next;
     ! Clock Cycle 1
     T = 1:
     next;
                                        ! Execute Assignment
     PHI1 = hi;
                                        ! Phase 1 Of
```

```
PHI2 = 10;
                               ! Clock Cycle 1
ASN = lo;
                               ! Assert Address Strobe
LIISN = 10;
                               ! Assert Lower Data Strobe
                               ! Assert Upper Data Strobe
UDSN = lo;
                               ! Enable Data Bus
DBENABLE = hi;
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 1
PHI2 = hi;
                               ! Execute Pending Assignments
next;
! Clock Cycle 2
T = 2;
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
FHI2 = 10;
                               ! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
                               ! Execute Impending Assignments
    next;
                               ! Phase 2
    fHI1 = lo;
                               ! Of Clock Cycle 2
    PHI2 = hi;
                               ! Execute Assignments
    next;
     T = 3;
                               ! Clock Cycle 3
    next;
                               ! Execute Assignment
    PHI1 = hi;
                               ! Phase 1
    PHI2 = 10;
                               ! Of Clock Cycle 3
    DBUS<15:8> = MEARUSD;
                               ! Memory Places Instruction
     DBUS<7:0> = MCABUS + 1];
                               ! On Data Bus And
    ITACKN = 10;
                               ! Asserts DTACKN(Added)
    next;
                               ! Execute Pending Assignments
     ! Return To Phase 2
     T = 2
                               ! Of Clock Cycle 2
    );
                               ! Execute Impending Assignments
    next;
! Clock Cycle 3
T = 3;
                               ! Execute Assignment
next;
                               ! Phase 2
FHI1 = 10;
                               ! Of Clock Cycle 3
PHI2 = hi;
                               ! Instruction On Data Bus
EXDRUF = DBUS;
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
```

```
T = 4;
                               ! Clock Cycle 4
next:
                               ! Execute Assignment
                               ! Phase 1
PHI1 = hi;
PHI2 = 10:
                               ! Of Clock Cycle 4
                               ! Store Displacement
DISREG = EXDBUF<7:0> sxt 32;
                               ! Execute Fending Assignments
next;
PHI1 = lo;
                               ! Phose 2
PHI2 = hi;
                               ! Of Clock Cycle 4
ASN = hi;
                               ! Deactivate Address Strobe
                               ! Deactivate Lower Data Strobe
LIISN = hi;
                               ! Deactivate Upper Data Strobe
UDSN = hi;
                               ! Are Flaced Into Instruction
                               ! Register
PC = PC + 2;
                               ! Increment Program Counter
DTACKN = hi;
                               ! Deactivate Data Transfer(Added)
                               ! Acknowledge
DISREG = DISREG + A[1];
                               ! Add AC13 To Displacement Register
next:
! Clock Cycle 5
next:
                               ! Execute Previous Assignment
PHI1 = hi;
                              ! Phase 1 Of
FHI2 = 10;
                               ! Clock Cycle 5
ABUS = 0xffffff;
                             ! Address Bus High Impedanced
DBUS = Oxffff:
                               ! Data Bus High Impedanced
                            ! Add Data Register To Displacement
DISREG = DISREG + DC73;
                               ! Execute Fending Assignments
next;
                               ! Phase 2 Of
PHI1 = lo;
                               ! Clock Cycle 5
PHI2 = hi;
                               ! Into External Address Buffer
next;
T = 6;
                               ! Clock Cycle 6
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1 Of
PHI2 = 10;
                               ! Clock Cycle 6
next;
                               ! Execute Pending Assignments
PHI1 = 10:
                               ! Phase 2
                               ! Of Clock Cycle 6
PHI2 = hi;
                               ! Execute Pending Assignments
T = 7;
                               ! Clock Cycle 7
next;
                               ! Execute Previous Assignment
```

```
FHI1 = hi;
                                ! Phase 1 Of
                                ! Clock Cycle 7
PHI2 = 10;
RW = hi;
                                ! Memory Read
ADENABLE = lo;
                                ! Disable Address Bus Buffer
                                ! Disable Data Bus Buffer
DEBENABLE = 10;
                                ! Place PC On Internal Address
IABUS<31:1> = PC<31:1>;
                                ! Bus
                                ! Execute Pending Assignments
next;
FHI1 = 10;
                                ! Phase 2 Of
                                ! Clock Cycle 7
PHI2 = hi;
                                ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                ! User Mode
FCSPACE = 2;
                                ! Accessing Data
EXABUF = IABUS<23:1>;
                                ! Gate Internal Address Bus
ABUS = IABUS<23:1>;
                                ! Place Address On Bus
next;
                                ! Into External Address Buffer
! Clock Cycle 8
T = 8;
                                ! Execute Assignment
next;
                                 ! Phase 1 Of
PHI1 = hi;
                                 ! Clock Cycle 8
PHI2 = 10;
                                ! Activate Upper And
UDSN = lo;
                                ! Lower Data Strobes
LDSN = lo;
ASN = lo;
                                ! Assert Address Strobe
                                ! Enable Data Bus
IJBENARLE = hi;
                                ! Execute Pending Assignments
next;
                                ! Phase 2
PHI1 = lo;
                                ! Of Clock Cycle B
PHI2 = hi;
                                 ! Execute Pending Assignments
next;
T = 9;
                                 ! Clock Cycle 9
                                 ! Execute Assignment
next;
                                ! Phase 1 Of
PHI1 = hi;
                                ! Clock Cycle 9
PHI2 = 10;
                                ! Wait For Memory To Place
while DTACKN eql hi
                                ! Ilata On The Bus
     next;
                                ! Execute Impending Assignments
                                ! Phase 2
     PHI1 = lo;
     PHI2 = hi;
                                ! Of Clock Cycle 9
     next;
                                 ! Execute Assignments
     ! Clock Cycle 10
     T = 10;
     next;
                                 ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
```

```
! Of Clock Cycle 10
     PHI2 = 10;
     DBUS<15:8> = M[ABUS];
                                ! Memory Places Instruction
    IDBUS<7:0> = MEABUS + 13;
                               ! On Data Rus And
                               ! Asserts DTACKN(Added)
    DTACKN = 10;
    next;
                                ! Execute Pending Assignments
     ! Return To Phase 2
                               ! Of Clock Cycle 9
     );
    next;
                               ! Execute impending Assignments
T = 10;
                                ! Clock Cycle 10
next;
                               ! Execute Assignment
PHI) = 10;
                               ! Phase 2
                               ! Of Clock Cycle 10
PHI2 = hi:
EXBRUF = DAUS:
                                ! Instruction On Nata Bus
                               ! Is Placed In External Data
                               ! kus Buffer
next;
                               ! Execute Pending Assignments
T = 11;
                                ! Clock Cycle 11
next;
                                ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                                ! Of Clock Cycle 11
PHI2 = 10;
                                ! The Contents Of The External
PFR = EXDBUF;
                                ! Data Rus Buffer Are Flaced
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 11
                               ! Deactivate Address Strobe
ASN = hi;
                                ! Neactivate Lower Data Strobe
LUSH = hi;
                                ! Decctivate Upper Data Strobe
UDSN = bli
                                ! Deactivate Data Transfer(Added)
DYACKE = his
                               ! Acknowledge
PC = PC + 2;
                               ! Increment PC
next:
                               ! Execute Pending Assignments
! Clock Cycle 12
T = 12;
next;
                                ! Execute Assignment
PHI1 = hi;
                               ! Phase 1 Of
                               ! Clock Cycle 12
PHI2 = 10;
RW = hi;
                               ! Memory Read
ADENABLE = 10;
                               ! Disable Address Bus Buffer
```

```
ABUS = Oxffffff;
                                  ! Address Bus High Impedanced
DBUS = 0xffff;
                                  ! Data Bus Returned To High
                                  ! Impedance State
                                  ! Disable Data Bus Buffer
DRENABLE = 10;
IABUS = DISREG:
                                  ! Place DISREG On Internal Address
                                  ! Execute Pending Assignments
next;
PHI1 = lo;
                                  ! Phase 2 Of
PHI2 = hi:
                                  ! Clock Cycle 12
                                  ! Enable Address Bus Buffer
ADENABLE = hi;
                                 ! User Mode
FCMODE = SKMODE:
                                 ! Accessing Data
FCSPACE = 1;
                                 ! Gate Internal Address Bus
EXABUF = IABUS<23:1>;
ARUS = IABUS<23:1>;
                                 ! Place Address On Bus
SRCARRY = 0;
                                  ! Initialize Status Register
                                  ! Condition Bits
SROVER = 0;
SRZERO = 0;
SRNEG = 0;
next:
                                 ! Into External Address Buffer
! Clock Cycle 13
T = 13;
next:
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1 Of
PHI2 = 16:
                                  ! Clock Cycle 13
                                  ! Activate Upper And
UDSN = 10;
LUSN = lo:
                                  ! Lower Data Strobes
ASN = lo:
                                  ! Assert Address Strobe
DBENABLE = hi;
                                  ! Enable Data Bus
                                  ! Execute Pending Assignments
next:
PHI1 = 10;
                                  ! Phase 2
FHI2 = hi;
                                  ! Of Clock Cycle 13
                                  ! Execute Pending Assignments
next;
! Clock Cycle 14
T = 14;
next;
                                  ! Execute Assignment
                                  ! Phase 1 Of
PHI1 = hi:
PHI2 = 10;
                                  ! Clock Cycle 14
                                  ! Wait For Memory To Place
while DTACKN eql hi
                                  ! Data On The Bus
                                  ! Execute Impending Assignments
     next;
     PHI1 = lo;
                                  ! Phase 2
     PHI2 = hi;
                                  ! Of Clock Cycle 14
                                  ! Execute Assignments
     next;
```

NEW PROPERTY OF THE PROPERTY O

```
/*************************************
                                 ! Clock Cycle 15
     T = 15;
                                 ! Execute Assignment
     next;
     PHI1 = hi;
                                ! Phase 1
     PHI2 = 10:
                                ! Of Clock Cycle 15
     DBUS<15:8> = MEABUSD;
                                ! Memory Places Instruction
     DBUS<7:0> = MCABUS + 13;
                                ! On Data Bus And
     DITACKN = 10;
                                ! Asserts DTACKN(Added)
     next;
                                 ! Execute Pending Assignments
     T = 14
                                 ! Return To Phase 2
                                 ! Of Clock Cycle 14
     );
                                ! Execute Impending Assignments
     next;
! Clock Cycle 15
T = 15;
next;
                                 ! Execute Assignment
                                ! Phase 2
PHI1 = lo;
PHI2 = hi:
                                ! Of Clock Cycle 15
EXDRUF = DBUS;
                                 ! Instruction On Data Bus
                                 ! Is Placed In External Data
                                 ! Bus Buffer
                                ! Execute Pending Assignments
next;
! Clock Cycle 16
T = 16;
                                 ! Execute Assignment
next;
PHI1 = hi;
                                ! Phase 1
                                 ! Of Clock Cycle 16
PHI2 = 10:
IDBUS = EXDBUF;
if EXDBUF eql 0
                                ! Set Condition Code Bits
  SRZERO = hi;
                                 ! As Appropriate
if EXDBUF<15>
  SRNEG = hi;
                                ! Execute Pending Assignments
next;
                                ! Phase 2
PHI1 = lo;
                                 ! Of Clock Cycle 16
FHI2 = hi;
                                 ! Place Value In Either
if IR eql 0x3431
                                 ! DE23 Or DE33
  D[2] = IDBUS
                                 ! Depending On Instruction
  D(3) = IDBUS;
                                 ! Deactivate Address Strobe
ASN = hi;
                                 ! Deactivate Lower Data Strobe
LDSN = hi;
                                 ! Deactivate Upper Data Strobe
UDSN = hi;
                                 ! Deactivate Data Transfer(Added)
DTACKN = hi;
                                 ! Acknowledge
```

IR = PFR;

```
! Execute Pending Assignments
     next;
     T = 0
                                     ! JMP (A0)
jmp :=
     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                      ! Clock Cycle 0
     DBUS = 0xffff;
                                      ! Place Data Bus In A High Impedance
     RW = hi;
                                      ! Memory Read
     AUENABLE = 10;
                                      ! Disable Address Bus Buffer
     DIBENABLE = 10;
                                      ! Disable Data Bus Buffer
                                      ! Place PC On Internal Address
     IABUS = PC;
                                      Bus
     next;
                                      ! Execute Pending Assignments
     PHI1 = lo;
                                      ! Phase 2 Of
                                      ! Clock Cycle 0
     PHI2 = hi;
                                      ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                      ! Gate Internal Address Bus
                                      ! Into External Address Buffer
                                      ! User Mode
     FCMODE = SRMODE;
     FCSPACE = 2;
                                      ! Accessing Program
                                      ! Execute Pending Assignments
     next;
     ABUS = EXABUF;
                                      ! Address Placed On Bus(Added)
     next;
                                      ! Execute Pending Assignments
     ! Clock Cycle 1
     T = 1;
                                      ! Execute Assignment
     next;
     PHI1 = hi;
                                      ! Phase 1 Of
     PHI2 = 10:
                                      ! Clock Cycle 1
     ASN = 10;
                                      ! Assert Address Strobe
     LDSN = lo:
                                      ! Assert Lower Data Strobe
     UDSN = lo;
                                      ! Assert Upper Data Strobe
     IABUS = A[0];
                                      ! Move Jump Address From A[O]
                                      ! To Internal Address Buffer
     DBENABLE = hi;
                                      ! Enable Data Bus
                                      ! Execute Pending Assignments
     next;
     PHI1 = lo;
                                      ! Phase 2
                                      ! Of Clock Cycle 1
     PHI2 = hi;
     PC = IABUS;
                                      ! Place Jump Address Into Program
                                      ! Counter
     next;
     T = 2;
                                      ! Clock Cycle 2
```

```
! Execute Assignment
next:
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
PHI2 = 10:
                              ! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Rus
                               ! Execute Impending Assignments
    next;
     FHI1 = 10;
                               ! Phase 2
    PHI2 = hi;
                               ! Of Clock Cycle 2
    next;
                               ! Execute Assignments
     7 = 3:
                               ! Clock Cycle 3
     next;
                               ! Execute Assignment
                               ! Phase 1
     PHI1 = hi;
                               ! Of Clock Cycle 3
     PHI2 = 10;
     DBUS<15:8> = MEABUS];
                               ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
     DITACKN = 10;
                               ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
     next;
     ! Return To Phase 2
     T = 2
                               ! Of Clock Cycle 2
     );
     next;
                               ! Execute Impending Assignments
T = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
                               ! Phase 2
PHI1 = lo:
                               ! Of Clock Cycle 3
PHI2 = hi:
EXDRUF = DRUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Rus Buffer
                               ! Execute Pending Assignments
next;
! Clock Cycle 4
T = 4:
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 4
next;
FFR = EXDBUF;
                               ! The Contents Of The External
                               ! Nata Bus Buffer Are Placed
                               ! In Prefetch Register
                               ! Execute Pending Assignments
next;
                               ! Phase 2
PHI1 = lo;
```

```
! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                                 ! Deactivate Address Strobe
LDSN = hi:
                                 ! Deactivate Lower Data Strobe
                                 ! Deactivate Upper Data Strobe
UDSN = hi:
DTACKN = hi;
                                 ! Deactivate Data Transfer
                                 ! Acknowledge(Added)
next:
! Clock Cycle 5
                                 ! Execute Previous Assignment
next;
PHI1 = hi:
                                 ! Phase 1 Of
                                 ! Clock Cycle 5
PHI2 = 10;
RW = hi;
                                 ! Memory Read
AUENABLE = 10:
                                 ! Disable Address Bus Buffer
DRENABLE = 10;
                                 ! Disable Data Bus Buffer
IABUS = PC;
                                 ! Place PC On Internal Address
                                 ! Execute Pending Assignments
next;
                                 ! Phase 2 Of
PHI1 = 10;
                                 ! Clock Cycle 5
PHI2 = hi:
                                 ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                 ! User Mode
                                 ! Accessing Program
FCSPACE = 2;
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
next;
                                 ! Into External Address Buffer
ABUS = EXABUF;
                                 ! Address Placed On Bus(Added)
next:
                                 ! Execute Pending Assignments
T = 6;
                                 ! Clock Cycle 6
                                 ! Execute Assignment
next;
                                 ! Phase 1 Of
PHI1 = hi:
FHI2 = 10;
                                 ! Clock Cycle 6
ASN = 10;
                                 ! Assert Address Strobe
LISN = lo;
                                 ! Assert Lower Data Strobe
UDSN = lo;
                                 ! Assert Upper Data Strobe
DENABLE = hi;
                                 ! Enable Data Rus
next;
                                 ! Execute Pending Assignments
PHI1 = 10:
                                 ! Phase 2
PHI2 = hi:
                                 ! Of Clock Cycle 6
next;
                                 ! Execute Pending Assignments
T = 7;
                                 ! Clock Cycle 7
next;
                                 ! Execute Assignment
PHI1 = hi:
                                 ! Phase 1
PHI2 = 10:
                                 ! Of Clock Cycle 7
while DTACKN eql hi
                                 ! Wait For Memory To Place
```

```
! Data On The Bus
    next:
                               ! Execute Impending Assignments
                               ! Phase 2
    PHI1 = lo;
    PHI2 = hi;
                               ! Of Clock Cycle 7
                               ! Execute Assignments
    next;
     T = 8:
                               ! Clock Cycle 8
                               ! Execute Assignment
    next;
    PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 8
    PHI2 = 10:
                               ! Memory Places Instruction
    DBUS<15:8> = MEABUSD;
                               ! On Data Rus And
    DBUS<7:0> = MEABUS + 13;
    DTACKN = lo:
                               ! Asserts BTACKN(Added)
                               ! Execute Fending Assignments
    next;
     ! Return To Phase 2
                              ! Of Clock Cycle 7
    );
    next;
                              ! Execute Impending Assignments
T = 8:
                               ! Clock Cycle 8
                               ! Execute Assignment
next;
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 8
PHI2 = hi;
                               ! Instruction On Data Bus
EXDBUF = DBUS;
                               ! Is Placed In External Data
                               ! Bus Ruffer
                               ! Execute Pending Assignments
next;
T = 9;
                               ! Clock Cycle 9
                               ! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi;
                               ! Of Clock Cycle 9
PHI2 = 10;
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 9
ASN = hi;
                               ! Deactivate Address Strobe
LDSN = hi;
                               ! Deactivate Lower Data Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
                               ! Increment Program Counter
FC = PC + 2;
IR = PFR;
                               ! Place Contents Of Prefetch
```

```
! Register Into Instruction
                                             ! Register
     DTACKN = hi;
                                            ! Deactivate Data Transfer
                                            ! Acknowledge(Added)
                                            ! Execute Pending Assignments
     next;
     T = 0
                                             ! Reset Clock Cycle Counter
      )
decode_execute_prefetch :=
                        case IR
                             0x3431,0x3631; move ! MOVE.W 4(A1,D7),D2 [D3]
                             0x027c; andi ! AND.W #$DFFF,SR
                                           ! JMP (AO) If IR = Octal Value
                             047320: jmp
                        esuc
                        )
main :=
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
           decode_execute_prefetch
     )
```

```
/*
                                               */
/*
    MOTOROLA MC68000 MODEL OF THE MOVE.W $2004,05 INSTRUCTION
                                               */
/*
/*
                                               */
/x
              Structure Declarations
                                               */
/*
                                               x/
state
/*
                                               */
/*
           Mó8000 Frogramming Registers
                                               */
/*
                                               */
DE0:73<31:0>.
                  ! 8 Data Registers
AE0:63<31:0>,
                  ! 7 Address Registers
UA7<31:0>,
                  ! User Stack Pointer
SA7<31:0>,
                  ! System Stack Pointer
PC<31:0>,
                  ! Program Counter
SR<15:0>,
                  ! Status Register
/*
                                               */
/*
                                               */
           Temporary Internal Registers
/*
                                               */
PFR<15:0>,
                  ! Prefetch Register
IR<15:0>,
                  ! Instruction Register
FC<2:0>,
                  ! Function Code Register
EXDBUF<15:0>,
                  ! External Data Bus Buffer Register
EXABUF<23:1>,
                  ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>.
                  ! ALU Buffer 1
ALUBUF2<31:0>.
                  ! ALU Buffer 2
DTEMP<15:0>.
                  ! Temporary Data Storage
DISREG<31:0>.
                  ! Temporary Displacement Storage
SRTEMP(15:0>,
                  ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPADR<31:0>,
                  ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>,
                  ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>.
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:0>,
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
T<7:0>.
                        ! Clock Cycle Counter
RESET.
                      ! Reset Flip-Flop
HALT,
                      ! Halt Flip-Flop
RW,
                     ! Read/Write Flip-Flop
ADENABLE.
                     ! Address Rus Buffer Enable
DIBENABLE,
                      ! Nata Bus Buffer Enable
ASN,
                      ! Address Strobe Flip-Flop
LIISN.
                      ! Lower Data Strobe Flip-Flop
UDSN.
                     ! Upper Data Strobe Flip-Flop
DITACKN.
                      ! Data Transfer Acknowledge Flip-Flop
COUT,
                      ! Carry Flip-Flop
EXCEPT,
                      ! Exception Processing Flip-Flop
READY,
                      ! Ready Flip-Flop
/*
                                                              */
/*
       Model transformation modifications:
                                                              */
/*
                                                              */
/*
           1) CDL decoder structure nonexistent in ISP' and un-
                                                              */
/*
       necessary for model. Eliminated.
                                                              */
/x
           Multi-phase clock structure nonexistent in ISP'.
                                                              */
/*
       Operations on registers will provide its equivalent.
                                                              ¥/
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              1/
/*
       register will provide its equivalent.
                                                              */
/*
           4) The declared bus structures are modeled with registers $/
       without loss of model accurracy. This done to maintain model
/*
                                                              */
/*
       equivalency and simplicity.
                                                              */
/*
           5) The memory word length was reduced from 16 to 8 bit
                                                              1/1
/*
       words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/*
       their PC incrementation, and to enable the use of existing
                                                              */
/*
       MC68000 assembler and linker/loader models. The memory was
                                                              1/
/*
       also reduced from 8 Mwords to 32 Kbytes.
                                                              */
/*
                                                              */
IABUS<31:0>,
                        ! Internal Address Bus
IDBUS<31:0>,
                        ! Internal Data Bus
                        ! Wait State Counter
twait<4:0>,
SWITCH,
                      ! Power Switch
PHI1,
                      ! Phase 1 Of Two-Phase Clock
PH12;
                      ! Phase 2 Of Two-Phase Clock
port
/*
                                                              */
/*
              External Address and Data Bus
                                                              */
/*
                                                              1/
DBUS<15:0>,
                        ! External Nata Rus
```

こうちょう かいしゅう アンプランド 一角 かかか かかい かいしゅう なんかん かんしゅう

```
ABUS<23:1>;
                     ! External Address Bus(changed)
formut
/#
/*
                Register Subfields
                                                        */
/*
                                                        */
PCAUDR
         = PC(23:0),
                      ! Frogram Counter Address Field
SRTRACE
         = SR(15).
                      ! Trace Bit
SKMODE
         = SR<13>.
                      ! Mode Selection Rit
         = Sik<0>,
SRCARRY
                      ! Carry Bit
SKOVER
         = SR<1>.
                      ! Overflow Rit
SRZERO
         = SR<2>.
                      ! Zero Bit
SRNEG
         = SR<3>,
                      ! Negative Bit
         = SR<4>,
SREX
                      ! Extend Bit
SRMASK
         = SR<10:8>,
                      ! Interrupt Mask
FCSFACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCMODE
                      ! User/Supervisor Mode Bit
         = FC(2).
F'CLOW
         = PC<15:0>.
                      ! PC Low Word
PCHI
                      ! PC High Word
         = PC(31115),
FIOL WORD
         = DE03<15:0>.
                      ! DEO3 Low Word
DILWORD
         = DE13<15:0>,
                      ! DE13 Low Word
D2LWORD
                      ! D[2] Low Word
         = D[2]<15:0>,
D3LWORD
         = DE3]<15:0>,
                      ! D[3] Law Word
L'4LWORL
                      ! D[4] Low Word
         = DE43<15:0>,
05LWORD
         = DC53<15:0>,
                      ! D(5) Low Word
D&LWORD
         = DE63<15:0>,
                      ! DE63 Low Word
D7LWORD
         = DC73<15:0>,
                      ! BE73 Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW
         = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
         = TEMPADR<31:16>;! TEMPADR High Word
TEMPADRHI
meaory
*/
/*
                                                        */
                 16K 16-Bit Word Internal Memory
                                                        1/
/*
ME0:327673<7:0>;
mac ro
/*
/*
                Logic Level Macros
                                                        */
```

```
/*
= 0 %,
10
hi
    = 1 &,
off
    = 0 %,
On
    = 1 %,
clear = 0 %;
/* Power On and Initialization. This process was not modeled but is
                                                       */
                                                       */
  added to initialize signals and registers.
/*
                                                       */
power_on_initialize :=
      SWITCH = on;
                               ! Turn Fower On
                                ! Execute Assignment
      next:
      READY = lo:
                                ! System Not Ready
      RESET = lo:
                                ! Assert Reset For
      delay(100);
                                ! 100 Miliseconds(Active Low)
      RESET = hi:
                                ! Deactivate Reset
      next;
                                ! Execute Pending Assignments
      ASN = hi;
                                ! Initialize Address Strobe
      LIISN = hi;
                                ! Initialize Lower Bata Strobe
      UDSN = hi;
                               ! Initialize Upper Data Strobe
      DITACKN = hi;
                               ! Initialize Nata Transfer Acknowledge
      RW = hi;
                               ! Initialize Read/Write(Read On High)
      DBUS = Oxffff:
                              ! Place Data Bus In High Impedance State
      M[0x100e] = 0xff;
                                ! Place Memory Locations Following The
      M[0x100f] = 0xff;
                                 ! JMP Instruction In A High State
      HALT = hi;
                                ! Initialize Halt Flip-Flop(Active
                                ! Low)
      T = 0;
                               ! Initialize Clock Cycle Counter
      READY = hi;
                                ! System Ready
      /*
                                                       */
                                                       */
           Routine Initialization fer Hamby and Guillory
      /*
      /*
                                                       */
      M[0x2004] = 0x55;
                                 ! Data To Re Moved
      ME0 \times 2005] = 0 \times 55;
      A[0] = 0 \times 1004;
                                ! Place Hex 1004 Into A[0]
      PC = 0x1000;
                                ! Place Hex 1000 Into Program Counter
      next
                                ! Execute Assignments
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                       */
```

/* to retrieve modeled instructions for simulation and analysis. It

```
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/* on page VI-15 of their thesis.
fetch_initial_instruction :=
     PHI1 = hi;
                                    ! Phase 1 Of
    PHI2 = 10;
                                    ! Clock Cycle 0
    RW = hi;
                                    ! Memory Read
     ADENABLE = 10;
                                    ! Disable Address Bus Buffer
    DRENABLE = 10;
                                    ! Disable Data Bus Buffer
                                    ! Place PC On Internal Address
    IARUS = PC;
    next;
                                    ! Execute Pending Assignments
    PHI1 = 10;
                                    ! Phase 2 Of
    PHI2 = hi:
                                    ! Clock Cycle 0
     ADENABLE = hi:
                                    ! Enable Address Bus Buffer
    EXABUF = IABUS;
                                    ! Gate Internal Address Bus
                                    ! Into External Address Buffer
    FCMODE = SRMODE:
                                    ! User Mode
    FCSPACE = 2;
                                    ! Accessing Program
    next:
                                    ! Execute Impending Assignments
    ARUS = EXABUF;
                                    ! Address Placed On Bus(Added)
                                     ! Execute Pending Assignments
     next;
     T = 1;
                                    ! Clock Cycle 1
    next;
                                    ! Execute Assignment
     PHI1 = hi;
                                    ! Phase 1 Of
    PHI2 = 10;
                                    ! Clock Cycle 1
     ASN = lo;
                                    ! Assert Address Strobe
    LDSN = lo;
                                    ! Assett Lower Data Strobe
     UDSN = 10:
                                    ! Assert Upper Data Strobe
     DBENABLE = hi:
                                    ! Enable Data Bus
     next;
                                    ! Execute Pending Assignments
     PHI1 = lo;
                                    ! Phase 2
     PHI2 = hi;
                                    ! Of Clock Cycle 1
                                    ! Execute Pending Assignments
     next;
     T = 2;
                                    ! Clock Cycle 2
     next;
                                     ! Execute Assignment
     PHI1 = h1;
                                    ! Phase 1
     PHI2 = 10:
                                    ! Of Clock Cycle 2
     while DTACKN eql hi
                                    ! Wait For Memory To Place
```

```
! Data On The Bus
    next;
                               ! Execute Impending Assignments
    PHI1 = 10;
                               ! Phase 2
    PHI2 = hi;
                               ! Of Clack Cycle 2
                               ! Execute Assignments
    next;
    T = 3:
                               ! Clock Cycle 3
    next;
                               ! Execute Assignment
    PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 3
    PHI2 = 10;
    DBUS<15:8> = MEABUSD;
                               ! Memory Places Instruction
    I(RUS<7:0) = MEARUS + 13;
                               ! On Data Bus And
    DTACKN = lo;
                               ! Asserts DTACKN(Added)
    next;
                               ! Execute Pending Assignments
     ! Return To Phase 2
                               ! Of Clock Cycle 2
    );
    next;
                               ! Execute Impending Assignments
T = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
FHI1 = lo;
                               ! Phase 2
PHI2 = hi:
                               ! Of Clock Cycle 3
                               ! Instruction On Data Bus
EXDBUF = DBUS;
                               ! Is Placed In External Data
                               ! Bus Buffer
next;
                               ! Execute Pending Assignments
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
                               ! Phase 1
PHI1 = hi;
                               ! Of Clock Cycle 4
PHI2 = 10;
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
                               ! Phase 2
PHI1 = 10;
                               ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                               ! Deactivate Address Strobe
                               ! Deactivate Lower Data Strobe
LDSN = hi;
UDSN = hi:
                               ! Deactivate Upper Data Strobe
                               ! Contents Of Prefetch Register
IR = PFR;
```

! Are flaced Into Instruction

```
! Register
     DTACKN = hi;
                                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
     PC = PC + 4;
                                        ! Increment Program Counter
     next:
                                        ! Execute Pending Assignments
     T = 0
                                        ! Reset Clock Cycle Counter
andi :=
                                        ! AND .W ##DFFF,SR
    SRMODE = 10;
                                        ! Effect Of Instruction
    IR<15:8> = MEPC1:
                                        ! Prefetch Next Instruction
    18<7:0> = MEPC + 13:
    next:
                                        ! Is To Set Status Register
       PC = PC + 2;
                                          ! Increment Program Counter
    T = 5;
                                        ! Supervisor Bit To User
    next;
                                        ! Mode
    T = 0
                                        ! Requires 6 Clock Cycles
move :=
                                        ! MOVE.W $2004,D5 [D6]
     PHI1 = hi;
                                        ! Phase 1 Of
                                        ! Clock Cycle 0
     PHI2 = 10;
     DBUS = Oxffff;
                                        ! Place Data Bus In High Impedance
     RW = hi;
                                        ! Memory Read
     ADENABLE = 10;
                                       ! Disable Address Bus Buffer
     ABUS = 0xffffff;
                                       ! Address Bus High Impedanced
     DBENABLE = 10;
                                       ! Disable Data Bus Buffer
     IABUS<31:1> = PC<31:1>;
                                        ! Place PC On Internal Address
                                        ! Bus
     next;
                                        ! Execute Pending Assignments
                                     · ! Phase 2 Of
     fHI1 = lo;
     PHI2 = hi;
                                       ! Clock Cycle 0
     ADENABLE = hi;
                                       ! Enable Address Bus Buffer
     EXABUF = IABUS<23:1>;
                                       ! Gate Internal Address Bus
                                       ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSFACE = 2:
                                        ! Accessing Program
     ABUS = IABUS<23;1>;
                                        ! Address Placed On Bus
                                        ! Execute Impending Assignments
     next:
     T = 1:
                                        ! Clock Cycle 1
     next;
                                        ! Execute Assignment
     PHI1 = hi;
                                        ! Phase 1 Of
     PHI2 = 10;
                                        ! Clock Cycle 1
     ASN = 10;
                                        ! Assert Address Strobe
```

```
LDSN = lo;
                              ! Assert Lower Data Strobe
UDSN = lo:
                              ! Assert Upper Data Strobe
DRENABLE = hi;
                              ! Enable Data Bus
next;
                              ! Execute Pending Assignments
                             ! Phase 2
PHI1 = lo;
                              ! Of Clock Cycle 1
PHI2 = hi:
                             ! Execute Pending Assignments
next;
! Clock Cycle 2
next;
                              ! Execute Assignment
FHI1 = hi;
                             ! Phase 1
PHI2 = 10:
                             ! Of Clock Cycle 2
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Data On The Bus
                             ! Execute Impending Assignments
    next:
    PHI1 = 10;
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 2
    next;
                              ! Execute Assignments
    T = 3;
                              ! Clock Cycle 3
    next;
                              ! Execute Assignment
    PHII = hi;
                              ! Phase 1
    PHI2 = 10;
                              ! Of Clock Cycle 3
    DBUS<15:8> = MEARUSD:
                              ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                              ! On Data Bus And
    INTACKN = 10;
                              ! Asserts DTACKN(Added)
    next:
                              ! Execute Pending Assignments
    T = 2
                              ! Return To Phase 2
                              ! Of Clock Cycle 2
    );
                              ! Execute Impending Assignments
    next;
T = 3:
                              ! Clock Cycle 3
next;
                              ! Execute Assignment
FHI1 = 10;
                              ! Phase 2
                              ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                              ! Instruction On Nata Bus
                              ! Is Placed In External Data
                              ! Bus Ruffer
next;
                              ! Execute Pending Assignments
T = 4;
                             ! Clock Cycle 4
```

```
next;
                                  ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
PHI2 = 10;
                                 ! Of Clock Cycle 4
DISREG = EXDBUF sxt 32;
                                 ! Store Displacement
next:
                                  ! Execute Pending Assignments
PHI1 = 10;
                                  ! Phase 2
PHI2 = hi;
                                  ! Of Clock Cycle 4
ASN = hi;
                                 ! Deactivate Address Strobe
LDSN = hi;
                                 ! Deactivate Lower Data Strobe
UDSN = hi:
                                  ! Deactivate Upper Data Strobe
                                  ! Are Fluced Into Instruction
                                  ! Register
PC = PC + 2;
                                  ! Increment Program Counter
DTACKN = hi;
                                  ! Deactivate Data Transfer(Added)
                                  ! Acknowledge
next;
T = 5;
                                  ! Clock Cycle 5
next;
                                  ! Execute Previous Assignment
PHI1 = hi;
                                  ! Phase 1 Of
                                  ! Clock Cycle 5
PHI2 = 10;
                                 ! Memory Read
RW = hi;
                                 ! Disable Address Bus Buffer
ADENABLE = lo;
                                 ! Address Bus High Impedanced
ABUS = 0xffffff;
DBUS = 0xffff;
                                 ! Data Bus Returned To High
                                 ! Impedance State
                                  ! Nisable Data Bus Buffer
DBENABLE = 10;
IABUS<31:1> = PC<31:1>;
                                  ! Flace FC On Internal Address
                                  ! Rus
next:
                                  ! Execute Pending Assignments
PHI1 = lo;
                                 ! Phase 2 Of
PHI2 = hi;
                                 ! Clock Cycle 5
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2;
                                 ! Accessing Data
EXABUF = IABUS(23:1);
                                 ! Gate Internal Address Bus
ABUS = IABUS<23:1>;
                                 ! Place Address On Bus
next;
                                 ! Into External Address Buffer
T = 6;
                                 ! Clock Cycle 6
next;
                                  ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10;
                                  ! Clock Cycle 6
UDSN = 10;
                                  ! Activate Upper And
LDSN = lo;
                                  ! Lower Data Strobes
                                  ! Assert Address Strobe
ASN = 10;
```

```
DBENABLE = hi;
                             ! Enable Data Rus
next;
                             ! Execute Pending Assignments
FHI1 = 10;
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 6
                             ! Execute Pending Assignments
next;
T = 7:
                             ! Clock Cycle 7
next:
                             ! Execute Assignment
PHI1 = hi;
                             ! Phase 1 Of
PHI2 = 10;
                             ! Clock Cycle 7
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Data On The Bus
    next:
                             ! Execute Impending Assignments
                             ! Phase 2
    PHI1 = 10:
    PHI2 = hi;
                             ! Of Clock Cycle 7
    next;
                            ! Execute Assignments
    T = 8:
                             ! Clock Cycle 8
    next;
                             ! Execute Assignment
    PHI1 = hi;
                             ! Phase 1
    PHI2 = 10;
                             ! Of Clock Cycle 8
    DBUS<15:8> = MEABUS];
                            ! Memory Places Instruction
    INBUS<7:0> = MEARUS + 13;
                            ! On Data Bus And
    DTACKN = lo;
                             ! Asserts DTACKN(Added)
    next;
                             ! Execute Pending Assignments
    ! Return To Phase 2
    T = 7
                             ! Of Clock Cycle 7
    );
                             ! Execute Impending Assignments
    next;
T = 8:
                             ! Clock Cycle 8
                             ! Execute Assignment
next;
PHI1 = 10;
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 8
EXBRUF = DRUS:
                             ! Instruction On Data Bus
                             ! Is Placed In External Data
                             ! Bus Buffer
next;
                             ! Execute Pending Assignments
T = 9:
                             ! Clock Cycle 9
next;
                             ! Execute Assignment
```

```
! Phase 1
PHI1 = hi;
PHI2 = 10:
                                   ! Of Clock Cycle 9
PFR = EXDBUF;
                                   ! The Contents Of The External
                                   ! Data Bus Buffer Are Placed
                                   ! In Prefetch Register
next;
                                   ! Execute Pending Assignments
PHI1 = lo;
                                   ! Phase 2
                                   ! Of Clock Cycle 9
PHI2 = hi;
ASN = hi;
                                   ! Deactivate Address Strobe
LDSN = hi;
                                   ! Deactivate Lower Data Strobe
UDSN = hi:
                                   ! Deactivate Upper Data Strobe
DTACKN = hi;
                                   ! Deactivate Data Transfer(Added)
                                   ! Acknowledge
PC = PC + 2;
                                   ! Increment PC
next;
                                   ! Execute Pending Assignments
T = 10;
                                    ! Clock Cycle 10
next;
                                   ! Execute Assignment
                                   ! Phase 1 Of
PHI1 = hi;
PHI2 = 10:
                                   ! Clock Cycle 10
F_{iW} = hi;
                                   ! Memory Read
ADENABLE = 10;
                                  ! Disable Address Bus Buffer
ABUS = 0xffffff;
                                  ! Address Bus High Impedanced
                                  ! Data Bus Returned To High
DBUS = 0xffff;
                                   ! Impedance State
DBENABLE = 10;
                                   ! Disable Data Bus Buffer
TABUS = DISREG;
                                     ! Place DISREG On Address
                                   ! Bus
                                   ! Execute Fending Assignments
next;
PHI1 = lo:
                                   ! Phase 2 Of
PHI2 = hi:
                                   ! Clock Cycle 10
ADENABLE = hi;
                                   ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                  ! User Mode
FCSPACE = 1;
                                  ! Accessing Data
EXABUF = IABUS<23:1>;
                                  ! Gate Internal Address Bus
ABUS = IABUS<23:1>;
                                   ! Place Address On Bus
SRCARRY = 0;
                                   ! Initialize Status Register
SROVER = 0;
                                   ! Condition Bits
SRZERO = 0;
SRNEG = 0;
                                   ! Into External Address Buffer
next;
T = 11;
                                    ! Clock Cycle 11
next;
                                   ! Execute Assignment
PHI1 = hi:
                                   ! Phase 1 Of
PHI2 = 10
                                   ! Clock Cycle 11
UUSN = lo;
                                   ! Activate Upper And
```

```
LDSN = 10;
                             ! Lower Data Strobes
ASN = 10;
                             ! Assert Address Strobe
DBENABLE = hi:
                             ! Enable Data Bus
next:
                             ! Execute Fending Assignments
PHI1 = lo;
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 11
                             ! Execute Pending Assignments
next;
! Clock Cycle 12:
T = 12;
                             ! Execute Assignment
next;
PHI1 = hi;
                             ! Phase 1 Of
PHI2 = 10;
                             ! Clock Cycle 12
                             ! Wait For Memory To Place
while DTACKN eql hi
                             ! Data On The Bus
                             ! Execute Impending Assignments
    next;
    FHI1 = lo;
                             ! Phase 2
    PHI2 = hi;
                            ! Of Clock Cycle 12
    next:
                             ! Execute Assignments
    T = 13:
                             ! Clock Cycle 13
    next;
                             ! Execute Assignment
    FHI1 = hi;
                             ! Phase 1
    PHI2 = 16;
                             ! Of Clock Cycle 13
    DBUS<15:8> = MCABUS];
                            ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                            ! On Data Bus And
    DITACKN = lo:
                             ! Asserts DTACKN(Added)
    next:
                             ! Execute Pending Assignments
    T = 12
                             ! Return To Phase 2
                             ! Of Clock Cycle 12
    );
    next;
                             ! Execute Impending Assignments
T = 13;
                             ! Clock Cycle 13
next;
                             ! Execute Assignment
                             ! Phase 2
PHI1 = lo;
                             ! Of Clock Cycle 13
PHI2 = hi;
EXDBUF = DBUS;
                             ! Instruction On Data Bus
                             ! Is Placed In External Data
                             ! Bus Buffer
                             ! Execute Pending Assignments
next:
```

! Clock Cycle 14

T = 14;

```
next;
                                           ! Execute Assignment
     PHI1 = hi;
                                           ! Phase 1
     PHI2 = 10;
                                           ! Of Clock Cycle 14
     IDBUS = EXDBUF;
     if EXDBUF eal O
                                           ! Set Condition Code Bits
         SRZERO = hi;
                                           ! As Appropriate
     if EXDBUF<15>
        SRNEG = hi:
     next;
                                           ! Execute Pending Assignments
     PHI1 = lo;
                                           ! Phase 2
     PHI2 = hi;
                                           ! Of Clock Cycle 14
     if IR eql 0x3u38
                                           ! Place Value In Either
         D(5) = IDRUS
                                           ! DC53 Or DC63
     else
                                           ! Depending On Instruction
        D[6] = IDBUS;
     ASN = hi:
                                           ! Deactivate Address Strobe
     LDSN = hi;
                                           ! Deactivate Lower Data Strobe
     UDSN = hi;
                                           ! Deactivate Upper Data Strobe
     DTACKN = hi;
                                           ! Deactivate Nota Transfer(Added)
                                           ! Acknowledge
     IR = PFR;
     next;
                                           ! Execute Pending Assignments
     T = 0
≃: برورز
                                           ! JMP (A0)
     PHI1 = hi;
                                           ! Phase 1 Of
     FHI2 = 10;
                                           ! Clock Cycle 0
     DBUS = 0xffff;
                                           ! Place Data Bus In A High Impedance
     RW = hi:
                                           ! Memory Read
     ADENABLE = 10;
                                           ! Disable Address Bus Buffer
     DBENABLE = lo:
                                           ! Disable Data Rus Buffer
     IARUS = PC:
                                           ! Place PC On Internal Address
                                           ! Bus
     next;
                                           ! Execute Pending Assignments
                                           ! Phase 2 Of
     PHI1 = 10;
     PHI2 = hi;
                                           ! Clock Eycle 0
     ADENABLE = hi;
                                           ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                           ! Gate Internal Address Bus
                                           ! Into External Address Buffer
     FCMODE = SRMODE:
                                           ! User Mode
     FCSFACE = 2;
                                           ! Accessing Program
     next;
                                           ! Execute Pending Assignments
     ARUS = EXABUF;
                                           ! Address Placed On Bus(Added)
      next;
                                           ! Execute Pending Assignments
```

```
T = 1;
                             ! Clock Cycle 1
next;
                             ! Execute Assignment
                             ! Phase 1 Of
PHI1 = hi;
                             ! Clock Cycle 1
PHI2 = 10;
ASN = lo;
                             ! Assert Address Strobe
LDSN = lo;
                             ! Assert Lower Data Strobe
                             ! Assert Upper Data Strobe
UDSN = lo;
IABUS = A[0];
                             ! Move Jump Address From ACOJ
                             ! To Internal Address Buffer
DBENABLE = hi;
                             ! Enable Data Bus
                             ! Execute Fending Assignments
next:
PHI1 = lo:
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 1
                             ! Place Jump Address Into Program
PC = IABUS;
                              ! Counter
next;
T = 2;
                             ! Clock Cycle 2
next:
                              ! Execute Assignment
PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 2
PHI2 = 10;
                             ! Wait For Memory To Place
while DTACKN eql hi
    (
                             ! Nata On The Bus
    next;
                             ! Execute Impending Assignments
    FHI1 = 10;
                            ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 2
                             ! Execute Assignments
    next:
    /**************************/
                            ! Clock Cycle 3
    T = 3;
    next;
                             ! Execute Assignment
    PHI1 = hi;
                             ! Phase 1
    PHI2 = 10;
                             ! Of Clock Cycle 3
    DBUS<15:8> = MCABUSI;
                            ! Memory Places Instruction
                            ! On Data Bus And
    DBUS<7:0> = MCABUS + 1];
                            ! Asserts DTACKN(Added)
    DITACKN = lo;
                             ! Execute Pending Assignments
    next:
    ! Return To Phase 2
    T = 2
                              ! Of Clock Cycle 2
    );
    next;
                          ! Execute Impending Assignments
```

```
T = 3:
                                   ! Clock Cycle 3
                                   ! Execute Assignment
next;
FHI1 = 10
                                   ! Phase 2
PHI2 = hi;
                                   ! Of Clock Cycle 3
EXDBUF = DRUS:
                                   ! Instruction On Nata Bus
                                   ! Is Placed In External Data
                                   ! Bus Buffer
next;
                                   ! Execute Pending Assignments
T = 4;
                                   ! Clock Cycle 4
next;
                                   ! Execute Assignment
PHI1 = hi;
                                   ! Phase 1
PHI2 = 10;
                                   ! Of Clock Cycle 4
next;
PFR = EXDBUF;
                                   ! The Contents Of The External
                                   ! Data Bus Buffer Are Placed
                                   ! In Prefetch Register
next;
                                   ! Execute Pending Assignments
FHI1 = 10;
                                   ! Phase 2
                                   ! Of Clock Cycle 4
PHI2 = hi:
ASN = hi;
                                   ! Deactivate Address Strobe
LUSN = hi;
                                   ! Deactivate Lower Data Strobe
UIISN = hi;
                                   ! Deactivate Upper Data Strobe
DTACKN = hi;
                                   ! Deactivate Nata Transfer
                                   ! Acknowledge(Added)
T = 5;
                                   ! Clock Cycle 5
                                   ! Execute Previous Assignment
next;
PHI1 = hi;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 5
RW = hi;
                                   ! Mesory Read
ADENABLE = 10;
                                   ! Disable Address Bus Buffer
DRENABLE = 10:
                                   ! Disable Data Bus Buffer
IARUS = PC;
                                   ! Place PC On Internal Address
                                   ! Bus
                                   ! Execute Pending Assignments
next;
                                   ! Phase 2 Of
PHI1 = lo;
                                   ! Clock Cycle 5
PHI2 = hi;
ADENABLE = hi;
                                   ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                   ! User Mode
FCSPACE = 2;
                                   ! Accessing Program
EXABUF = IABUS;
                                   ! Gate Internal Address Bus
next;
                                   ! Into External Address Buffer
ABUS = EXABUF;
                                   ! Address Placed On Bus(Added)
                                   ! Execute Pending Assignments
next;
```

```
T = 6;
                             ! Clock Cycle 6
next;
                             ! Execute Assignment
PHI1 = hi;
                             ! Phase 1 Of
                             ! Clock Cycle 6
PHI2 = 10:
                             ! Assert Address Strobe
ASN = 10;
                             ! Assert Lower Data Strobe
LISN = lo;
                             ! Assert Upper Data Strobe
UDSN = lo;
DBENABLE = hi;
                             ! Enable Data Bus
next;
                             ! Execute Pending Assignments
PHI1 = lo:
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 6
                             ! Execute Pending Assignments
next;
T = 7;
                             ! Clock Cycle 7
                             ! Execute Assignment
next;
PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 7
PHI2 = 10;
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Data On The Bus
                             ! Execute Impending Assignments
    next;
    PHI1 = 10;
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 7
    next;
                             ! Execute Assignments
    T = 8;
                             ! Clock Cycle 8
    next;
                             ! Execute Assignment
    PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 8
    PHI2 = 10;
    DBUS<15:8> = MCABUS3:
                             ! Memory Places Instruction
    IBUS<7:0> = MCABUS + 13;
                             ! On Data Bus And
    ITACKN = 10:
                             ! Asserts BTACKN(Added)
    next:
                             ! Execute Pending Assignments
    T = 7
                             ! Return To Phase 2
                             ! Of Clock Cycle 7
    );
                            ! Execute Impending Assignments
    next;
T = 8:
                             ! Clock Cycle 8
next;
                             ! Execute Assignment
PHI1 = lo;
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 8
```

```
EXDBUF = DBUS:
                                          ! Instruction On Data Bus
                                          ! Is Placed In External Data
                                          ! Bus Buffer
                                          ! Execute Pending Assignments
     next:
     T = 9;
                                          ! Clock Cycle 9
     next;
                                          ! Execute Assignment
     PHI1 = hi;
                                          ! Phase 1
     FHI2 = 10;
                                          ! Of Clock Cycle 9
     FFR = EXDBUF;
                                          ! The Contents Of The External
                                          ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
                                          ! Execute Pending Assignments
     next;
     PHI1 = lo;
                                          ! Phase 2
                                          ! Of Clock Cycle 9
     PHI2 = hi:
     ASN = hi;
                                          ! Deactivate Address Strobe
     LDSN = hi;
                                          ! Deactivate Lower Data Strobe
     UDSN = hi;
                                          ! Deactivate Upper Data Strobe
     PC = PC + 2;
                                          ! Increment Program Counter
     IR = PFR;
                                          ! Place Contents Of Prefetch
                                          ! Register Into Instruction
                                          ! Register
     DTACKN = hi;
                                          ! Deactivate Data Transfer
                                          ! Acknowledge(Added)
                                          ! Execute Pending Assignments
     next;
     T = 0
                                          ! Reset Clock Cycle Counter
decode_execute_prefetch :=
                       case IR
                           0x3a38,0x3c38; move
                                               ! MOVE.W $2004.R5 [D6]
                           0x027c; andi ! AND.W #$DFFF,SR
                           047320: jmp
                                          ! JMP (AO) If IR = Octal Value
                       esuc
                       )
main :=
    power_on_initialize;
    fetch_initial_instruction;
    while READY eql hi
          decode_execute_prefetch
    )
```

```
/*
    MOTORGLA MC68000 MODEL OF THE MOVE.W $2004,$2008 INSTRUCTION
/*
                                                */
/*
                                                */
/*
                                                */
/*
              Structure Declarations
                                                */
/*
                                                */
state
/*
                                                */
/*
           M68000 Programming Registers
                                                */
/x
                                                */
DE0:73<31:0>,
                  ! 8 Data Registers
AC0:63<31:0>,
                  ! 7 Address Registers
UA7<31:0>.
                  ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Pointer
PC<31:0>,
                   ! Program Counter
SR<15:0>,
                   ! Status Register
/*
                                                */
                                                */
/¥
           Temporary Internal Registers
                                                */
/*
PFR<15:0>,
                   ! Prefetch Register
                   ! Instruction Register
IR<15:0>,
FC<2:0>,
                   ! Function Code Register
EXDBUF<15:0>,
                   ! External Data Bus Buffer Register
EXABUF<23:1>,
                   ! External Address Bus Buffer Register(changed)
                  ! ALU Buffer 1
ALUBUF1<31:0>,
ALUBUF2<31:0>,
                  .! ALU Buffer 2
DTEMP<15:0>.
                   ! Temporary Data Storage
DISREG<31:0>,
                   ! Temporary Displacement Storage
SRTEMP<15:0>,
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
                   ! Temporary Instruction Register Storage
IRTEMP<15:0>,
                   ! (Exception Processing)
TEMPADR<31:0>,
                   ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>,
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>,
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

. .

```
! Temporary Address Storage For
HANADR<31:0>,
                        ! Exception Handler Routine
T<7:0>,
                        ! Clock Cycle Counter
                     ! Reset Flip-Flop
RESET.
HALT.
                     ! Halt Flip-Flop
RW.
                     ! Read/Write Flip-Flop
                     ! Address Bus Buffer Enable
ADENABLE,
IIBENABLE,
                     ! Data Bus Buffer Enable
                     ! Address Strobe Flip-Flop
ASN,
                     ! Lower Nata Strobe Flip-Flop
LIISN,
                     ! Upper Data Strobe Flip-Flop
UDISN,
DITACKN,
                     ! Nata Transfer Acknowledge Flip-Flop
COUT,
                     ! Carry Flip-Flop
EXCEPT,
                     ! Exception Processing Flip-Flop
READY,
                     ! Ready Flip-Flop
/*
                                                              */
/*
       Model transformation modifications:
                                                              */
/*
                                                              */
           1) CDL decoder structure nonexistent in ISP' and un-
/*
                                                              */
/*
       necessary for model. Eliminated.
                                                              */
/*
           2) Multi-phase clock structure nonexistent in ISP'.
                                                              */
/*
       Operations on registers will provide its equivalent.
                                                              */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              */
/x
       register will provide its equivalent.
                                                              1/
/*
           4) The declared bus structures are modeled with registers */
/x
       without loss of model accurracy. This done to maintain model
                                                              */
18
       equivalency and simplicity.
                                                              */
/×
           5) The memory word length was reduced from 16 to 8 bit
                                                              */
       words to coincide with the ECB's 32-Kbyte memory, to agree with*/
/x
/x
       their PC incrementation, and to enable the use of existing
                                                             1/
/*
       MC68000 assembler and linker/loader models. The memory was
                                                              */
/*
       also reduced from 8 hwords to 32 Kbytes.
                                                              */
                                                              */
/*
IABUS<31:0>,
                        ! Internal Address Bus
IDBUS<31:0>,
                        ! Internal Data Bus
                        ! Wait Cycle Counter
twait<7:0>,
SWITCH,
                     ! Power Switch
                     ! Phase 1 Of Two-Phase Clock
PHI1,
                     ! Phase 2 Of Two-Phase Clock
PHI2;
port
*/
/*
/*
             External Address and Data Bus
                                                              */
/x
                                                              */
DBUS<15:0>,
                       ! External Data Bus
```

```
ABUS<23:1>;
                      ! External Address Bus(changed)
format
*/
/*
/*
                                                         */
                Register Subfields
/*
                                                         */
FCADOR
         = PC<23:0>.
                      ! Program Counter Address Field
SRTRACE
         = SR(15),
                      ! Trace Nit
SAMODE
         = SR<13>,
                      ! Mode Selection Bit
SRCARRY
         = SR<0>,
                      ! Carry Bit
SROVER
                      ! Overflow Bit
         = SR<1>,
SRZERO
         = SR(2),
                       Zero Bit
SRNEG
         = SR<3>,
                      ! Negative Bit
SREX
                      ! Extend Bit
         = SR<4>,
SRMASK
         = SR<10:8>.
                      ! Interrupt Mask
FCSPACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCMODE
         = FC(2),
                      ! User/Supervisor Mode Bit
PCLOW
         = PC<15:0>.
                      ! PC Low Word
                      ! PC High Word
PCHI
         = PC<31:16>,
DOLWORD
         = D[0](15:0).
                      ! IIEOJ Low Word
DILWORD
         = BC13<15:0>.
                      ! DE13 Low Word
D2LWORD
         = DE23<15:0>,
                      ! D[2] Low Word
D3LWORD
         = DC33<15:0>,
                      ! DE31 Low Word
                      ! BE43 Low Word
II4LWORI
         = BC43<15:0>,
DSLWORD
                      ! DC53 Low Word
         = D(5)(15:0),
II6LWORI
         -= DE63<15:0>,
                      ! II[6] Low Word
D7LWORD
         = DC73<15:0>,
                      ! DE73 Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADIRLOW
         = HANADR<15:00, ! HANADR Low Word
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADREGW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI
         = TEMPADR<31:16>;! TEMPADR High Word
DIEDOTY
*/
/*
                                                         */
/*
                 16K 16-Bit Word Internal Memory
11
ME0:327673<7:0>;
BIGCTO
/*
/*
                Logic Level Macros
                                                         */
```

おいいいいんには はつべんがいいち まっているいない

```
/x
10
    = 01,
hi
    = 1 &,
off
    = 0 &,
on
    = 1 &,
clear = 0 %;
*/
/*
/* Power On and Initialization. This process was not modeled but is
                                                        */
                                                        x/
   added to initialize signals and registers.
power_on_initialize :=
                                ! Turn Power On
      SWITCH = on;
                                ! Execute Assignment
      next;
      REALIY = lo;
                                ! System Not Ready
                                ! Assert Reset For
      RESET = 10;
                                ! 100 Miliseconds(Active Low)
      delay(100);
                                ! Deactivate Reset
      RESET = hi:
                                ! Execute Pending Assignments
      next:
                                ! Initialize Address Strobe
      ASN = hi;
      LISN = hi;
                                ! Initialize Lower Data Strobe
      UDSN = hi:
                                ! Initialize Upper Data Strobe
      DITACKH = hi;
                                ! Initialize Nata Transfer Acknowledge
      RW = hi;
                                ! Initialize Read/Write(Read On High)
                                ! Place Data Bus In High Impedance State
      INBUS = Oxffff;
                                ! Place Memory Locations Following The
      MEO \times 10103 = O \times ff;
                                  ! JMP Instruction In A High State
      M[0x1011] = 0xff;
                                ! Initialize Halt Flip-Flop(Active
      HALT = hi;
                                ! Low)
                                ! Initialize Clock Cycle Counter
      T = 0;
      REARY = hi;
                                 ! System Ready
      /*
           Routine Initialization Per Hamby and Guillory
                                                        */
      /*
                                                        */
      M[0x2004] = 0x55;
                                ! Data To Be Moved
      ME0 \times 2005] = 0 \times 55;
      A[0] = 0 \times 1004;
                                ! Place Hex 1004 Into A[0]
      PC = 0 \times 1000;
                                ! Place Hex 1000 Into Program Counter
      next
                                 ! Execute Assignments
/x
                                                        1/
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                        */
/* to retrieve modeled instructions for simulation and analysis. It
```

```
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/* on page VI-15 of their thesis.
                                                           */
/*
                                                           */
fetch_initial_instruction :=
     FHI1 = hi;
                                     ! Phase 1 Of
     PHI2 = 10;
                                     ! Clock Cycle 0
     RW = hi;
                                     ! Memory Read
     ADENABLE = lo;
                                     ! Disable Address Bus Buffer
     DBENABLE = 10:
                                    ! Disable Data Bus Buffer
     IABUS = PC;
                                     ! Place PC On Internal Address
                                     ! Execute Pending Assignments
     next;
     FHI1 = 10;
                                     ! Phase 2 Of
     PHI2 = hi;
                                     ! Clock Cycle 0
     ADENABLE = hi;
                                    ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                    ! Gate Internal Address Bus
                                    ! Into External Address Buffer
     FCMODE = SRMODE;
                                    ! User Mode
     FCSPACE = 2;
                                    ! Accessing Program
                                    ! Execute Impending Assignments
     next;
     ABUS = EXABUF;
                                     ! Address Placed On Bus(Added)
     next;
                                     ! Execute Pending Assignments
     T = 1:
                                     ! Clock Cycle 1
     next;
                                    ! Execute Assignment
                                    ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10:
                                    ! Clock Cycle 1
    .ASN = 10;
                                    ! Assert Address Strobe
     LDSN = lo:
                                    ! Assert Lower Data Strobe
     UDSN = lo:
                                    ! Assert Upper Data Strobe
     DBENABLE = hi;
                                     ! Enable Data Bus
     next;
                                     ! Execute Pending Assignments
     PHI1 = lo;
                                     ! Phase 2
     PHI2 = hi;
                                     ! Of Clock Cycle 1
                                    ! Execute Pending Assignments
     next;
     T = 2:
                                    ! Clock Cycle 2
     next;
                                     ! Execute Assignment
                                     ! Phase 1
     PHI1 = hi:
                                     ! Of Clock Cycle 2
     PHI2 = 10;
     while DTACKN eql hi
                                     ! Wait For Memory To Place
```

```
! Data On The Bus
     next;
                                ! Execute Impending Assignments
    PHI1 = 10:
                                ! Phase 2
     PHI2 = hi;
                                ! Of Clock Cycle 2
                               ! Execute Assignments
     next;
     1 = 3:
                               ! Clock Cycle 3
     next;
                                ! Execute Assignment
    PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 3
     PHI2 = 10;
     DBUS<15:8> = MCABUS3;
                               ! Memory Places Instruction
     DBUS<7:0> = MCABUS + 13;
                               ! On Data Bus And
                                ! Asserts DTACKN(Added)
     DTACKN = 10;
                                ! Execute Pending Assignments
     next;
     T = 2
                                ! Keturn To Phase 2
                                ! Of Clock Cycle 2
     );
     next;
                                ! Execute Impending Assignments
T = 3;
                                ! Clock Cycle 3
next;
                                ! Execute Assignment
PHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 3
EXDRUF = DRUS;
                                ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Bus Buffer
                                ! Execute Pending Assignments
next;
T = 4;
                               ! Clock Cycle 4
                                ! Execute Assignment
next;
                                ! Phase 1
PHI1 = hi:
                                ! Of Clock Cycle 4
PHI2 = 10:
                                ! The Contents Of The External
PFR = EXDBUF;
                                ! Data Bus Buffer Are Placed
                                ! In Prefetch Register
next;
                                ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
                                ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi:
                                ! Deactivate Address Strobe
                                ! Deactivate Lower Data Strobe
LUSN = hi;
UDSN = hi;
                                ! Deactivate Upper Data Strobe
IR = PFR;
                                ! Contents Of Prefetch Register
                                ! Are Placed Into Instruction
```

```
! Register
     DTACKN = hi;
                                       ! Deactivate Data Transfer(Added)
                                       ! Acknowledge
     PC = PC + 4;
                                       ! Increment Program Counter
                                       ! Execute Pending Assignments
     next;
     T = 0
                                       ! Reset Clock Cycle Counter
     )
                                       ! MOVE.W $2004,$2008
BIOVE :=
     PHI1 = hi;
                                       ! Phase 1 Of
     PHI2 = 10;
                                       ! Clock Cycle 0
     DBUS = 0xfiff;
                                       ! Place Data Bus In High Impedance
                                       ! Memory Read
     RW = hi;
     ADENABLE = 10;
                                       ! Disable Address Bus Buffer
                                       ! Disable Data Bus Buffer
     DEBENABLE = 10;
     IABUS = PC;
                                       ! Place PC On Internal Address
                                       ! Rus
     next;
                                       ! Execute Pending Assignments
     PHI1 = lo:
                                       ! Phase 2 Of
     PHI2 = hi;
                                       ! Clock Cycle 0
     ADENABLE = hi;
                                       ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                       ! Gate Internal Address Bus
                                       ! Into External Address Buffer
     FCMODE = SRMODE;
                                       ! User Mode
                                       ! Accessing Program
     FCSFACE = 2;
     ABUS = IABUS;
                                       ! Address Placed On Bus(Added)
                                       ! Execute Pending Assignments
     next:
     T = 1:
                                       ! Clock Cycle 1
     next;
                                       ! Execute Assignment
     PHI1 = hi:
                                       ! Phase 1 Of
     PHI2 = 10;
                                       ! Clock Eycle 1
                                       ! Assert Address Strobe
     ASN = 10;
                                       ! Assert Lower Data Strobe
     LDSN = lo;
                                       ! Assert Upper Data Strobe
     UDSN = lo;
     DBENABLE = hi;
                                       ! Enable Data Bus
                                       ! Execute Fending Assignments
     next;
     PHI1 = 10;
                                       ! Phase 2
                                       ! Of Clock Cycle 1
     PHI2 = hi;
                                       ! Execute Pending Assignments
     next;
     T = 2;
                                       ! Clock Cycle 2
                                       ! Execute Assignment
     nexti
```

```
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
PHI2 = 10;
while DTACKN eql hi
                               ! Wait For Memory To Place
                               ! Data On The Bus
                               ! Execute Impending Assignments
    next;
                               ! Phase 2
     PHI1 = lo:
     PHI2 = hi;
                               ! Of Clock Cycle 2
                               ! Execute Assignments
     next;
     T = 3:
                               ! Clock Cycle 3
     next;
                               ! Execute Assignment
     PHI1 = hi;
                               ! Phase 1
     PHI2 = 10;
                               ! Of Clock Cycle 3
     DBUS<15:8> = MEABUS3;
                               ! Memory Places Instruction
     DBUS<7:0> = MCABUS + 13;
                               ! On Data Bus And
                               ! Asserts DTACKN(Added)
     DITACKN = 10;
    next;
                               ! Execute Pending Assignments
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
    );
     next;
                               ! Execute Impending Assignments
T = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DRUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
next;
                               ! Execute Pending Assignments
T = 4:
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 4
PHI2 = 10;
                               ! The Contents Of The External
TEMPADRHI = EXDBUF;
                               ! Data Bus Buffer Are Placed
                               ! In Temporary Address Register
                               ! Execute Pending Assignments
next;
PHI1 = 10:
                               ! Phase 2
                               ! Of Clock Cycle 4
PHI2 = hi:
ASN = hi;
                               ! Deactivate Address Strobe
                               ! Deactivate Lower Data Strobe
LDSN = hi;
```

```
UDSN = hi:
                                 ! Deactivate Upper Data Strobe
                                 ! Are Placed Into Instruction
                                 ! Register
DTACKN = hi;
                                 ! Deactivate Data Transfer(Added)
                                 ! Acknowledge
PC = PC + 2;
                                 ! Increment PC
next:
T = 5;
                                 ! Clock Cycle 5
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1 Of
                                 ! Clock Cycle 5
PHI2 = 10;
DBUS = 0xffff;
                                 ! Place Data Bus In High Impedance
RW = hi;
                                 ! Memory Read
ADENABLE = lo:
                                 ! Disable Address Bus Buffer
DRENABLE = lo:
                                 ! Disable Data Bus Buffer
IABUS = PC:
                                 ! Place PC On Internal Address
next;
                                 ! Execute Pending Assignments
PHI1 = lo;
                                 ! Phase 2 Of
                                 ! Clock Cycle 5
PHI2 = hi;
                                 ! Enable Address Rus Ruffer
ADENABLE = hi;
EXABUF = IABUS;
                                ! Gate Internal Address Bus
                                ! Into External Address Buffer
FCMODE = SRHODE;
                                ! User Mode
FCSPACE = 2:
                                 ! Accessing Program
ABUS = IABUS;
                                ! Address Placed On Bus(Added)
                                ! Execute Fending Assignments
nexti
T = 6;
                                 ! Clock Cycle 6
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1 Of
fHI2 = 10;
                                 ! Clock Cycle 6
ASN = 10:
                                 ! Assert Address Strobe
                                 ! Assert Lower Data Strobe
LUSN = 10;
UDSN = lo:
                                 ! Assert Upper Data Strobe
DRENABLE = hi;
                                 ! Enable Data Bus
next;
                                 ! Execute Pending Assignments
PHI1 = lo;
                                 ! Phase 2
PHI2 ≈ hi;
                                 ! Of Clock Cycle 6
                                 ! Execute Pending Assignments
next;
T = 7:
                                 ! Clock Cycle 7
                                 ! Execute Assignment
next;
```

```
PHI1 = hi;
                                ! Phase 1
PHI2 = 10;
                                ! Of Clock Cycle 7
while DTACKN eql hi
                                ! Wait For Memory To Place
                                ! Data On The Rus
     next;
                                ! Execute Impending Assignments
     FHI1 = lo;
                                ! Phase 2
                                ! Of Clock Cycle 7
     PHI2 = hi;
     next;
                                ! Execute Assignments
     ! Clock Cycle 8
     next;
                                ! Execute Assignment
     PHI1 = hi:
                                ! Phase 1
     PHI2 = 10;
                                ! Of Clock Cycle 8
     DBUS<15:8> = MCABUSD;
                                ! Memory Places Instruction
     DBUS<7:0> = MCABUS + 13;
                                ! On Data Bus And
                                ! Asserts LITACKN(Added)
     DITACKN = 10;
     next;
                                ! Execute Pending Assignments
     T = 7
                                ! Return To Phase 2
                                ! Of Clock Cycle 7
     );
     next;
                               ! Execute Impending Assignments
T = 8;
                                ! Clack Cycle 8
next:
                                ! Execute Assignment
PHI1 = lo:
                                ! Phase 2
PHI2 = hi:
                                ! Of Clock Cycle 8
EXDBUF = DBUS;
                                ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Bus Buffer
next;
                                ! Execute Pending Assignments
\z*********************************
T = 9
                                ! Clock Cycle 9
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1
PHI2 = 10;
                                ! Of Clock Cycle 9
                                ! The Contents Of The External
TEMPADRICU = EXDEUF;
                                ! Nata Bus Buffer Are Placed
                                ! In Temporary Register
next;
                                ! Execute Pending Assignments
PHI1 = lo:
                                ! Phase 2
PHI2 = hi:
                                ! Of Clock Cycle 9
ASN = hi;
                                ! Deactivate Address Strobe
LDSN = hi;
                                ! Deactivate Lower Data Strobe
```

```
UDSN = hi;
                                 ! Deactivate Upper Data Strobe
                                 ! Are Placed Into Instruction
                                 ! Register
DTACKN = hi;
                                 ! Deactivate Data Transfer(Added)
                                 ! Acknowledge
PC = PC + 2;
                                 ! Increment PC
next;
T = 10;
                                 ! Clock Cycle 10
next;
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10:
                                 ! Clock Cycle 10
                                 ! Place Data Bus In High Impedance
INBUS = Oxffff;
RW = hi;
                                 ! Memory Read
ADENABLE = 10;
                                 ! Disable Address Bus Ruffer
                                 ! Disable Data Bus Buffer
DBENABLE = 10;
IABUS = PC;
                                 ! Place PC On Internal Address
next;
                                 ! Execute Pending Assignments
PHI1 = 10;
                                 ! Phase 2 Of
PHI2 = hi;
                                 ! Clock Cycle 10
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
                                 ! Gate Internal Address Bus
EXABUF = IABUS:
                                 ! Into External Address Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2;
                                 ! Accessing Program
ABUS = IABUS;
                                ! Address Placed On Bus(Added)
                                 ! Execute Fending Assignments
next;
T = 11;
                                 ! Clock Cycle 11
                                 ! Execute Assignment
next;
PHI1 = hi:
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 11
ASN = 10;
                                 ! Assert Address Strobe
LDSN = 10;
                                 ! Assert Lower Data Strobe
                                 ! Assert Upper Data Strobe
UDSN = lo;
                                 ! Enable Data Bus
DBENABLE = hi;
next;
                                 ! Execute Pending Assignments
PHI1 = 10;
                                 ! Phase 2
PHI2 = hi:
                                 ! Of Clock Cycle 11
                                 ! Execute Pending Assignments
T = 12;
                                  ! Clock Cycle 12
                                 ! Execute Assignment
next;
```

というではないなからは、これのからののは、一方がなっ

```
PHI1 = hi;
                             ! Phase 1
                               ! Of Clock Cycle 12
FHI2 = 10;
while DTACKN eql hi
                               ! Wait For Memory To flace
                               ! Data On The Bus
                               ! Execute Impending Assignments
    next;
                            ! Phase 2
    PHI1 = 10;
                               ! Of Clock Cycle 12
    PHI2 = hi;
    next;
                               ! Execute Assignments
    ! Clock Cycle 13
    T = 13:
                               ! Execute Assignment
    next;
    PHI1 = hi;
                               ! Phase 1
    PHI2 = 10;
                               ! Of Clock Cycle 13
                               ! Memory Places Instruction
    DBUS<15:8> = MEABUS3;
     DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
    INTACKN = 10;
                               ! Asserts DTACKN(Added)
    next:
                               ! Execute Pending Assignments
     ! Return To Phase 2
    T = 12
                               ! Of Clock Cycle 12
    );
                               ! Execute Impending Assignments
    next:
! Clock Cycle 13
T = 13:
next;
                               ! Execute Assignment
FHI1 = 10;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 13
                               ! Instruction On Data Rus
EXDBUF = DBUS;
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
T = 14:
                               ! Clock Cycle 14
                               ! Execute Assignment
next;
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 14
                               ! Execute Pending Assignments
next;
                               ! Phase 2
PHI1 = lo;
                               ! Of Clock Cycle 14
PHI2 = hi;
                               ! Deactivate Address Strobe
ASN = hi;
                               ! Deactivate Lower Data Strobe
LDSN = hi;
UDSN = hi;
                               ! Deactivate Upper Data Strobe
                               ! Are Placed Into Instruction
                               ! Register
```

THE RESERVE WAS A STATE OF THE PARTY OF THE

```
DTACKN = hi;
                                  ! Deactivate Data Transfer(Added)
                                  ! Acknowledge
PC = PC + 2;
                                  ! Increment PC
next:
T = 15;
                                  ! Clock Cycle 15
next;
PHI1 = hi;
                                  ! Phase 1 Of
PHI2 = 10;
                                  ! Clock Cycle 15
INBUS = Oxffff;
                                  ! Place Data Bus In High Impedance
RW = hi;
                                  ! Memory Read
ADENABLE = 10;
                                  ! Disable Address Bus Buffer
DBENABLE = lo:
                                  ! Disable Data Bus Buffer
IABUS = TEMPADR:
                                  ! Place TEMPADE On Internal Address
next;
                                  ! Execute Pending Assignments
PHI1 = lo;
                                  ! Phase 2 Of
PHI2 = hi;
                                  ! Clock Cycle 15
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
                                 ! Into External Address Buffer
                                 ! User Mode
FCMODE = SRMODE;
FCSPACE = 1;
                                 ! Accessing Program
TEMPADRHI = EXDBUF;
                                 ! Store High Word Of Destination
                                ! Address Placed On Bus(Added)
ABUS = IABUS;
                                 ! Execute Pending Assignments
next:
T = 16;
                                  ! Clock Cycle 16
next;
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1 Of
PHI2 = 10;
                                  ! Clock Cycle 16
                                  ! Assert Address Strobe
ASN = 10;
LDSN = lo;
                                 ! Assert Lower Data Strobe
                                 ! Assert Upper Data Strobe
UDSN = lo;
DBENABLE = hi;
                                  ! Enable Data Bus
next;
                                  ! Execute Pending Assignments
PHI1 = 10;
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 16
next:
                                 ! Execute Pending Assignments
T = 17;
                                  ! Clock Cycle 17
                                  ! Execute Assignment
next;
PHI1 = hi;
                                  ! Phase 1
                                  ! Of Clock Cycle 17
PHI2 = 10;
```

```
! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
                               ! Execute Impending Assignments
    next;
    PHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 17
    PHI2 = hi;
                               ! Execute Assignments
    next;
    ! Clock Cycle 18
    next;
                               ! Execute Assignment
    FHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 18
    PHI2 = 10;
    DBUS<15:8> = MCARUSJ;
                               ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                              ! On Data Bus And
    DITACKN = 10;
                               ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
    next;
    ! Return To Phase 2
    T = 17
                               ! Of Clock Cycle 17
    );
    next;
                             ! Execute Impending Assignments
T = 18;
                               ! Clock Cycle 18
                               ! Execute hasignment
Nexti
                               ! Phase 2
FH11 = 10;
PHI2 = hi;
                               ! Of Clock Cycle 18
                               ! Instruction On Data Bus
EXDBUF = DBUS:
                               ! Is Placed In External Data
                               ! Rus Buffer
                               ! Execute Pending Assignments
next;
! Clock Cycle 19
T = 19;
                               ! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi;
                               ! Of Clock Cycle 19
PHI2 = 10
                               ! The Contents Of The External
DITEMP = EXDBUF;
                               ! Nata Bus Buffer Are Placed
                               ! In Temporary Register
                               ! Execute Pending Assignments
next;
                               ! Phase 2
FHI1 = 10;
                               ! Of Clock Cycle 19
PHI2 = hi;
                               ! Deactivate Address Strobe
ASN = hi;
                               ! Deactivate Lower Data Strobe
LDSN = hi;
                               ! Deactivate Upper Data Strobe
UDSN = hi;
                               ! Are Placed Into Instruction
```

```
! Register
                                 ! Deactivate Data Transfer(Added)
DTACKN = hi;
                                 ! Acknowledge
next;
T = 20;
                                 ! Clock Cycle 20
next;
                                 ! Phase 1 Of
PHI1 = hi:
                                 ! Clock Cycle 20
PHI2 = 10;
DBUS = Oxffff;
                                 ! Place Data Bus In High Impedance
RW = hi;
                                 ! Memory Read
                                 ! Disable Address Bus Buffer
ADENABLE = 10;
                                 ! Disable Data Bus Buffer
DBENABLE = 10;
IABUS = PC;
                                 ! Place PC On Internal Address
next:
                                 ! Execute Pending Assignments
PHI1 = lo;
                                 ! Phase 2 Of
PHI2 = hi:
                                 ! Clock Cycle 20
ADENABLE = hi:
                                 ! Enable Address Rus Ruffer
                                 ! Gate Internal Address Rus
EXABUF = IABUS;
                                 ! Into External Address Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2;
                                 ! Accessing Frogram
                                ! Address Placed On Bus(Added)
ABUS = IABUS;
                                 ! Execute Pending Assignments
next;
T = 21;
                                  ! Clock Cycle 21
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 21
ASN = 10;
                                 ! Assert Address Strobe
                                 ! Assert Lower Data Strobe
LDSN = lo;
UDSN = lo:
                                 ! Assert Upper Data Strobe
                                 ! Enable Data Bus
DBENABLE = hi;
                                 ! Execute Pending Assignments
next;
PHI1 = 10;
                                 ! Phase 2
PHI2 = hi:
                                 ! Of Clock Cycle 21
next;
                                 ! Execute Pending Assignments
T = 22;
                                  ! Clock Cycle 22
                                 ! Execute Assignment
next;
                                 ! Phase 1
PHI1 = hi;
PHI2 = 10:
                                 ! Of Clock Cycle 22
```

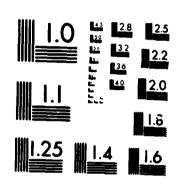
! Wait For Memory To Place

while DTACKN eql hi

```
! Nata On The Rus
     next;
                                ! Execute Impending Assignments
     PHI1 = 10:
                                ! Phase 2
     PHI2 = hi;
                                ! Of Clock Cycle 22
     next:
                                ! Execute Assignments
     T = 23;
                                ! Clock Cycle 23
                                ! Execute Assignment
     next;
     PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 23
     PHI2 := 10;
                                ! Memory Places Instruction
     DBUS<15:8> = MEABUS];
     DBUS<7:0> \approx MEABUS + 13;
                                ! On Data Bus And
     INTACKN = 10;
                                ! Asserts DTACKN(Added)
                                ! Execute Pending Assignments
     next;
     T = 22
                                 ! Return To Phase 2
                                ! Of Clock Cycle 22
     );
     next;
                                ! Execute Impending Assignments
T = 23;
                                ! Clock Cycle 23
next;
                                ! Execute Assignment
FH11 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 23
EXDBUF = DBUS;
                                ! Instruction On Data Rus
                                ! Is Placed In External Data
                                ! Bus Buffer
next;
                                ! Execute Pending Assignments
T = 24:
                                ! Clock Cycle 24
                                ! Execute Assignment
next:
PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 24
PHI2 = 10;
TEMPADRLOW = EXDBUF;
                                ! The Contents Of The External
                                ! Data Bus Buffer Are Flaced
                                ! In Temporary Register
                                ! Execute Pending Assignments
next;
PHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 24
                                ! Deactivate Address Strobe
ASN = hi;
                                ! Deactivate Lower Data Strobe
LDSN = hi;
UDSN = hi;
                                ! Deactivate Upper Data Strobe
                                ! Are Placed Into Instruction
                                ! Register
```

```
DTACKN = hi;
                                   ! Deactivate Data Transfer(Added)
                                   ! Acknowledge
PC = PC + 2;
                                   ! Increment PC
next;
T = 25;
                                    ! Clock Cycle 25
                                   ! Execute Frevious Assignment
next;
PHI1 = hi;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 25
RW = hi;
                                   ! Memory Read
ADENABLE = 10;
                                   ! Disable Address Bus Buffer
DBUS = Oxffff;
                                   ! Data Bus Returned To High
                                   ! Impedance State
DRENABLE = 10;
                                   ! Disable Data Bus Buffer
                                 ! Place TEMPADR On Internal Address
IABUS = TEMPADR;
                                   | Rus
                                   ! Execute Pending Assignments
next;
                                   ! Phase 2 Of
PHI1 = 10;
                                   ! Clock Cycle 25
PHI2 = hi;
                                   ! Enable Address Bus Buffer
ABENABLE = h1;
FCMODE = SRMODE;
                                   ! User Mode
                                   ! Accessing Program
FCSPACE = 1;
EXABUF = IABUS;
                                   ! Gate Internal Address Bus
                                   ! Place Low Word from DTEMP On
IDBUS = DTEMP;
                                   ! Internal Data Bus
                                  ! Address Placed On Bus(Added)
ABUS = 1ABUS;
                                   ! Execute Pending Assignments
next;
T = 26;
                                    ! Clock Cycle 26
next;
                                   ! Execute Assignment
                                   ! Phase 1 Of
PHI1 = hi;
                                   ! Clock Cycle 26
PHI2 = 10;
                                   ! Assert Address Strobe
ASN = 10;
RW = 10
                                   ! Place Contents Of Internal
EXDRUF = IBBUS;
                                   ! Data Bus Into External Data Buffer
SECARRY = 10;
                                   ! Reset Condition Code Bits
SROVER = 10;
SRZERO = lo;
SRNEG = 10;
next;
                                   ! Execute Pending Assignments
                                   ! Phase 2
PHI1 = lo;
                                   ! Of Clock Cycle 26
PHI2 = hi;
if EXDBUF eql 0
                                   ! Set Zero Condition Bit If Needed
  SRZERO = hi;
                                   ! Place Data On External Data Bus
DRUS = EXDRUF;
                                   ! Enable Data Bus
DBENABLE = hi;
```

THE SIMULATION AND AMALYSIS OF A RTL MODEL OF THE MOTOROLA MC68000 MICROP. (U) AIR FORCE INST OF TECH MRIGHT-PATTERSON AFB OH SCHOOL OF ENGI. C A BAXLEY DEC 84 AFIT/GCS/ENG/940-2-VOL-2 F/G 9/2 AD-A164 257 4/5 UNCLASSIFIED



MICROCOPY RESOLUTION TEST CHART

```
next;
                              ! Execute Pending Assignments
T = 27:
                              ! Clock Cycle 27
next;
                              ! Execute Assignment
PHI1 = hi;
                             ! Phase 1
                              ! Of Clock Cycle 27
PHI2 = 10;
                             ! Set Negative Condition Bit
if EXDRUF<15>
  SRNEG = hi;
                             ! If Needed
UDSN = lo;
                             ! Activate Upper And
LIISN = 10;
                             ! Lower Data Strobes
twait = 0;
                             ! Wait Cycle Counter Initialized
next;
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Data On The Rus
                             ! Increment Wait Cycle
    tweit = tweit + 1;
                             ! Execute Impending Assignments
    next;
   FHI1 = 10;
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 27
    next;
                              ! Execute Assignments
    /*********************************
    T = 28:
                              ! Clock Cycle 28
    next;
                             ! Execute Assignment
                             ! Phase 1
    PHI1 = hi;
                             ! Of Clock Cycle 28
    PHI2 = 10;
    if twait eql 2
                             ! Memory Responds After 2 Cycles
    MEARUSD = DBUS<15:8>:
                            ! Store Data From Bus
    MEABUS + 13 = DBUS<7:0>;
                             ! In Memory
                             ! Asserts DTACKN(Added)
    DITACKN = 10
    );
    next;
                             ! Execute Pending Assignments
    ! Return To Phase 2
    T = 27
                             ! Of Clock Cycle 27
    );
                            ! Execute Impending Assignments
    next;
T = 28;
                              ! Clock Cycle 28
next;
                              ! Execute Assignment
FHI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 28
PHI2 = hi;
                              ! Execute Pending Assignments
next;
1 = 29:
                              ! Clock Cycle 29
```

```
! Execute Assignment
next;
PHI1 = hi:
                                   ! Phase 1
                                   ! Of Clock Cycle 29
PHI2 = 10;
next;
                                   ! Execute Fending Assignments
PHI1 = lo;
                                   ! Phase 2
                                   ! Of Clock Cycle 29
PHI2 = hi;
ASN = hi;
                                   ! Deactivate Address Strabe
LDSN = hi;
                                   ! Deactivate Lower Nata Strobe
                                   ! Deactivate Upper Data Strobe
UDSN = hi;
IJTACKN = hi;
                                   ! Deactivate Data Transfer
                                   ! Acknowledge(Added)
                                   ! Execute Pending Assignments
next;
T = 30;
                                   ! Clock Cycle 30
next;
                                   ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                                   ! Clock Cycle 30
DBUS = 0xffff;
                                   ! Place Data Bus In High Impedance
RW = hi;
                                   ! Memory Read
ADENABLE = lo;
                                   ! Disable Address Bus Buffer
                                   ! Disable Data Bus Buffer
DRENABLE = 10;
                                   ! Place PC On Internal Address
IAKUS = PC:
next;
                                    ! Execute Pending Assignments
                                   ! Phase 2 Of
PHI1 = 10;
PHI2 = hi;
                                   ! Clock Cycle 30
ADENABLE = hi:
                                   ! Enable Address Bus Buffer
EXABUF = IABUS;
                                   ! Gate Internal Address Bus
                                 · ! Into External Address Buffer
FCMODE = SRMODE:
                                   ! User Mode
FCSPACE = 2;
                                   ! Accessing Program
ARUS = IABUS;
                                   ! Address Placed On Bus(Added)
next:
                                   ! Execute Pending Assignments
T = 31;
                                    ! Clock Cycle 31
next;
                                   ! Execute Assignment
                                   ! Phase 1 Of
PHI1 = hi;
                                   ! Clock Cycle 31
PHI2 = 10:
                                   ! Assert Address Strobe
ASN = lo:
                                   ! Assert Lower Data Strobe
LDSN = 10;
UI(SN = 10;
                                   ! Assert Upper Data Strobe
DBENABLE = hi;
                                   ! Enable Data Bus
next;
                                   ! Execute Pending Assignments
PHI1 = 10;
                                   ! Phase 2
```

THE CANADA STATE OF THE STATE O

```
! Of Clock Cycle 31
PHI2 = hi;
next;
                              ! Execute Pending Assignments
T = 32;
                              ! Clock Cycle 32
next;
                              ! Execute Assignment
                              ! Phase 1
PHI1 = hi;
PHI2 = 10;
                              ! Of Clock Cycle 32
                              ! Wait For Memory To Place
while DTACKN eql hi
                              ! Data On The Bus
                             ! Execute Impending Assignments
    next;
    PHI1 = lo;
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 32
                             ! Execute Assignments
    next;
    /**********************************
    T = 33:
                              ! Clock Cycle 33
                              ! Execute Assignment
    next;
                              ! Phase 1
    PHI1 = hi;
    PHI2 = 16;
                             ! Of Clock Cycle 33
    DBUS<15:8> = MEABUS3:
                             ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                             ! On Data Bus And
    DITACKN = 10;
                             ! Asserts DTACKN(Added)
    next:
                              ! Execute Pending Assignments
    T = 32
                              ! Return To Phase 2
                              ! Of Clock Cycle 32
    );
                             ! Execute Impending Assignments
    next;
T = 33:
                              ! Clock Cycle 33
                              ! Execute Assignment
next;
FHI1 = 10;
                              ! Phase 2
PHI2 = hi;
                              ! Of Clock Cycle 33
EXDBUF = DBUS;
                              ! Instruction On Data Bus
                              ! Is Placed In External Data
                              ! Bus Buffer
                              ! Execute Pending Assignments
next:
! Clock Cycle 34
T = 34;
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1
                              ! Of Clock Cycle 34
PHI2 = 10;
PFR = EXDBUF;
                              ! The Contents Of The External
                              ! Data Bus Buffer Are Placed
```

```
! In Prefetch Register
     next;
                                           ! Execute Fending Assignments
     FHI1 = 10;
                                           ! Phase 2
     PHI2 = hi;
                                           ! Of Clock Cycle 34
     ASN = hi;
                                           ! Deactivate Address Strobe
     LUSN = hi;
                                           ! Deactivate Lower Data Strobe
                                           ! Deactivate Upper Data Strobe
     UDSN = hi;
                                           ! Are Flaced Into Instruction
                                           ! Register
     LITACKN = hi:
                                           ! Deactivate Data Transfer(Added)
                                           ! Acknowledge
     IR = PFR;
                                           ! Load Instruction Register
                                           ! With Prefetch Register
     PC = PC + 2;
                                           ! Increment PC
     next;
     T = 0
andi :=
                                           ! AND.W ##DFFF,SR
    SRMODE = 10:
                                           ! Effect Of Instruction
    IR<15:8> = MEPC3;
                                           ! Prefetch Next Instruction
    1R<7:0> = MCPC + 13;
                                           ! Is To Set Status Register
    next;
       PC = PC + 2;
                                              ! Increment Program Counter
     T = 5;
                                           ! Supervisor Rit To User
                                           ! Mode
    next;
     T = 0
                                           ! Requires 6 Clock Cycles
    )
jap :=
                                           ! JMP (A0)
      ! Phase 1 Of
     PHI1 = hi;
     FHI2 = 10;
                                           ! Clock Cycle 0
                                           ! Place Data Bus In A High Impedance
     DBUS = 0xffff:
                                           ! Memory Read
     RW = hi;
                                           ! Disable Address Bus Buffer
     ADENABLE = 10;
                                           ! Disable Data Bus Buffer
     IIRENABLE = 10;
      IABUS = PC;
                                           ! Place PC On Internal Address
     next;
                                           ! Execute Pending Assignments
     PHI1 = 10;
                                           ! Phase 2 Of
                                           ! Clock Cycle 0
     PHI2 = hi;
                                           ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS:
                                           ! Gate Internal Address Bus
                                           ! Into External Address Buffer
     FCMODE = SRMODE;
                                           ! User Mode
```

```
FCSPACE = 2;
                                ! Accessing Program
next;
                                ! Execute Pending Assignments
ARUS = EXABUF;
                                ! Address Placed On Bus(Added)
                                ! Execute Pending Assignments
next:
T = 1:
                                ! Clock Cycle 1
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1 Of
                                ! Clock Cycle 1
PHI2 = 10;
                                ! Assert Address Strobe
ASN = lo;
                                ! Assert Lower Data Strobe
LDSN = lo;
                                ! Assert Upper Data Strobe
UDSN = lo;
1ABUS = A[0];
                                ! Move Jump Address From A[0]
                                ! To Internal Address Buffer
                                ! Enable Data Rus
DRENABLE = hi;
                                ! Execute Fending Assignments
next:
                                ! Phase 2
PHI1 = lo;
                                ! Of Clock Cycle 1
PHI2 = hi;
                                ! Place Jump Address Into Program
PC = IABUS:
                                 ! Counter
next;
! Clock Cycle 2
T = 2i
                                 ! Execute Assignment
next;
                                ! Phase 1
PHI1 = hi;
f'HI2 = 10;
                                ! Of Clock Cycle 2
while DTACKN eql hi
                                ! Wait For Memory To Place
                                ! Data On The Bus
                                ! Execute Impending Assignments
     next;
                                ! Phase 2
     FHI1 = lo;
                                ! Of Clock Cycle 2
     PHI2 = hi;
                                ! Execute Assignments
     next;
     T = 3;
                                ! Clock Cycle 3
                                ! Execute Assignment
     next;
     PHI1 = hi;
                                ! Phase 1
     PHI2 = 10:
                                ! Of Clock Cycle 3
                                ! Memory Places Instruction
     DBUS(15:8> = MEARUS];
                                ! On Data Bus And
     DBUS<7:0> = MCABUS + 13;
                                ! Asserts DTACKN(Added)
     DITACKN = 10;
                                 ! Execute Pending Assignments
     next;
                                ! Return To Phase 2
     T = 2
                                 ! Of Clock Cycle 2
```

```
);
     next:
                                 ! Execute Impending Assignments
T = 3:
                                 ! Clock Cycle 3
next;
                                 ! Execute Assignment
PHI1 = lo;
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 3
                                 ! Instruction On Data Bus
EXDBUF = DBUS;
                                 ! Is Placed In External Data
                                 ! Bus Buffer
                                 ! Execute Pending Assignments
next;
T = 4:
                                 ! Clock Cycle 4
                                 ! Execute Assignment
next;
PHI1 = hi;
                                 ! Phase 1
                                 ! Of Clock Cycle 4
FHI2 = 10;
next;
PFR = EXDBUF;
                                 ! The Contents Of The External
                                 ! Data Bus Buffer Are Placed
                                 ! In Prefetch Register
next;
                                 ! Execute Pending Assignments
FHI1 = lo;
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 4
ASN = hi;
                                 ! Deactivate Address Strobe
LDSN = hi;
                                 ! Deactivate Lower Data Strobe
                                 ! Neactivate Upper Nata Strobe
UIISN = hi;
                                 ! Deactivate Data Transfer
DIACKN = hi;
                                 ! Acknowledge(Added)
T = 5;
                                 ! Clock Cycle 5
                                 ! Execute Previous Assignment
next;
PHI1 = hi;
                                 ! Phase 1 Of
fHI2 = 10;
                                 ! Clock Cycle 5
RW = hi;
                                 ! Memory Read
                                 ! Disable Address Bus Buffer
ADENABLE = 10;
                                 ! Disable Data Bus Buffer
DRENABLE = 10;
IABUS = PC;
                                 ! Place FC On Internal Address
                                 ! Rus
next;
                                 ! Execute Pending Assignments
PHI1 = lo;
                                 ! Phase 2 Of
PHI2 = hi;
                                 ! Clock Cycle 5
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2;
                                 ! Accessing Program
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
```

```
! Into External Address Buffer
next;
ARUS = EXABUF;
                              ! Address Flaced On Rus(Added)
next:
                              ! Execute Pending Assignments
! Clock Cycle 6
T = 6;
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1 Of
FHI2 = 10;
                              ! Clock Cycle 6
                              ! Assert Address Strobe
ASN = 10;
LDSN = lo;
                              ! Assert Lower Data Strobe
UIISN = lo;
                              ! Assert Upper Data Strobe
DIBENABLE = hi;
                              ! Enable Data Rus
next:
                              ! Execute Pending Assignments
FHI1 = 10;
                              ! Phase 2
PHI2 = hi;
                              ! Of Clock Cycle 6
                              ! Execute fending Assignments
next;
T = 7;
                              ! Clock Cycle 7
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1
PHI2 = 10;
                              ! Of Clock Cycle 7
                              ! Wait For Memory To Flace
while DTACKN eql hi
                              ! Data On The Bus
    next;
                              ! Execute Impending Assignments
    PHI1 = 10;
                              ! Phase 2
    FHI2 = hi;
                              ! Of Clock Cycle 7
                              ! Execute Assignments
    next;
    T = 8:
                              ! Clock Cycle 8
    next;
                              ! Execute Assignment
    PHI1 = hi;
                              ! Phase 1
    PHI2 = 10;
                              ! Of Clock Cycle 8
    DBUS<15:8> = MCABUS3;
                              ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                              ! On Data Bus And
                              ! Asserts DTACKN(Added)
    DTACKN = lo;
    next;
                              ! Execute Fending Assignments
    T = 7
                             ! Return To Phase 2
                             ! Of Clock Cycle 7
    );
    next;
                             ! Execute Impending Assignments
T = 8;
                              ! Clock Cycle 8
```

```
! Execute Assignment
     next;
                                          ! Phase 2
     PHI1 = 10;
                                          ! Of Clock Cycle 8
     PHI2 = hi;
                                          ! Instruction On Data Rus
     EXDRUF = DRUS;
                                          ! Is Placed In External Data
                                          ! Bus Buffer
                                          ! Execute Pending Assignments
     next;
     T = 9;
                                          ! Clock Cycle 9
     next;
                                          ! Execute Assignment
     PHI1 = hi;
                                          ! Phase 1
                                          ! Of Clock Cycle 9
     PHI2 = 10;
                                          ! The Contents Of The External
     PFR = EXDBUF:
                                          ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
     next;
                                          ! Execute Pending Assignments
                                          ! Phase 2
     PHI1 = lo;
                                          ! Of Clock Cycle 9
     PHI2 = hi;
                                          ! Deactivate Address Strobe
     ASN = hi;
     LDSN = hi:
                                          ! Deactivate Lower Data Strobe
                                          ! Deactivate Upper Data Strobe
     UDSN = hi;
     PC = PC + 2;
                                          ! Increment Program Counter
     IR = PFR:
                                          ! Place Contents Of Prefetch
                                          ! Register Into Instruction
                                          ! Register
                                          ! Deactivate Data Transfer
     DTACKN = hi:
                                          ! Acknowledge(Added)
                                          ! Execute Pending Assignments
     next;
     T = 0
                                          ! Reset Clock Cycle Counter
decode_execute_prefetch :=
                       case IR
                            0x33f9: move
                                          ! MOVE.W $2004.$2008
                            047320: jap
                                          ! JMP (AO) If IR = Octal Value
                            0x027c: andi
                                          ! AND.W ##DFFF,SR
                       esac
                       )
main :=
    power_on_initialize;
    fetch_initial_instruction;
    while READY eql hi
          decode_execute_prefetch
    )
```

```
11
                                                x/
/*
    MOTORULA MC68000 MODEL OF THE MOVE.W #$5555, III INSTRUCTION
                                                */
/*
                                                */
/*
                                                */
/#
              Structure Declarations
                                                1/
/*
state
/*
                                                */
           Mó8000 Frogramming Registers
/*
                                                */
/x
                                                1/
DE0:73(31:0>,
                   ! 8 Data Registers
A[0:63<31:0>,
                  ! 7 Address Registers
UA7<31:0>,
                  ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Pointer
                   ! Program Counter
PC<31:0>,
CRK15:0>,
                   ! Status Register
/*
/*
           Temporary Internal Registers
                                                */
/*
PFR<15:0>,
                   ! Prefetch Register
                   ! Instruction Register
IR<15:0>,
EXÉRUPÇIS:0>,
                   ! Function Code Register
! External Data Bus Buffer Register
EXABUF<23:1>,
                   ! External Address Rus Buffer Register(changed)
ALUBUF1<31:0>,
                   ! ALU Buffer 1
                   ! ALU Buffer 2
ALUBUF2(31:0),
DITEMP<15:0>.
                   ! Temporary Data Storage
IdSREG<31:0>.
                   ! Temporary Displacement Storage
SRTEMP<15:0>,
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPADR<31:0>.
                   ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>,
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>,
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

これである。マングランは、東京のこれののは、東京のこれでは、

```
HANADR<31:0>,
                         ! Temporary Address Storage For
                          ! Exception Hundler Routine
                         ! Clock Cycle Counter
T<7:0>,
RESET,
                       ! Reset Flip-Flop
                       ! Halt Flip-Flop
HALT,
Ŕ₩,
                       ! Read/Write Flip-Flop
ADENABLE,
                       ! Address Bus Buffer Enable
DIBENABLE,
                       ! Nata Bus Buffer Enable
ASN,
                       ! Address Strobe Flip-Flop
LDSN,
                       ! Lower Nata Strobe Flip-Flop
ULISN,
                       ! Upper Data Strobe Flip-Flop
DITACKN.
                       ! Lata Transfer Acknowledge Flip-Flop
COUT,
                       ! Carry Flip-Flop
EXCEPT,
                       ! Exception Processing Flip-Flop
READY,
                       ! Ready Flip-Flop
/*
                                                                 */
/*
       Model transformation modifications:
                                                                 */
/*
                                                                 */
/x
            1) CDL decoder structure nonexistent in ISP' and un-
                                                                 */
       necessary for model. Eliminated.
/*
                                                                 */
/×

 Multi-phase clock structure nonexistent in ISP'.

                                                                 */
/*
       Operations on registers will provide its equivalent.
                                                                 */
/*
            3) Switch structure nonexistent in ISP'. Operation on a
                                                                 */
/×
                                                                 *. '
       register will provide its equivalent.
/*
            4) The declared bus structures are modeled with registers */
/*
       without loss of model accurracy. This done to maintain model
                                                                 ¥/
/×
                                                                 */
       equivalency and simplicity.
/*
            5) The memory word length was reduced from 16 to 8 bit
                                                                 */
/*
       words to coincide with the ECK's 32-Kbyte memory, to agree with*/
/ X
       their PC incrementation, and to enable the use of existing
                                                                 */
/*
       MC68000 assembler and linker/loader models. The memory was
                                                                 */
/x
       also reduced from 8 Mwords to 32 Kbytes.
                                                                 */
/*
                                                                 */
1ABUS<31:0>,
                          ! Internal Address Bus
IDBUS<31:0>,
                         ! Internal Data Bus
                          ! Wait Cycle Counter
twait<7:0>.
                       ! Power Switch
SWITCH,
                       ! Phase 1 Of Two-Phase Clock
FH11.
                       ! Phase 2 Of Two-Phase Clock
PHI2;
port
/*
                                                                 */
/*
              External Address and Data Bus
                                                                 */
/*
                                                                 */
! External Nata Bus
DBUS<15:0>,
```

アン・シングへご

```
ABUS<23:1>;
                      ! External Address Bus(changed)
format
1*
                                                         1/
/*
                Register Subfields
                                                         */
/*
                                                         */
PCADDR
         = PC<23:0>,
                      ! Program Counter Address Field
SRTRACE
         = SR<15>,
                      ! Trace Bit
SAMODE
         = SR<13>.
                      ! Mode Selection Bit
SRCARRY
         = S8<0>.
                      ! Carry Rit
         = SR<1>,
SROVER
                      ! Overflow Bit
SRZERO
         = SR<2>.
                      ! Zero Bit
SRNEG
         = SR(3),
                      ! Negative Bit
SREX
                      ! Extend Bit
         = SR<4>,
         = SR<10:8>,
SRMASK
                      ! Interrupt Mask
                      ! Memory Access Address Space
FCSFACE
         = FC<1:0>,
FCMODE
         = FC<2>,
                      ! User/Supervisor Mode Bit
F'CLOW
                      ! PC Low Word
         = PC<15:0>,
PCHI
                      ! FC High Word
         = PC(31:16),
DOLMORD
         = D[0]<15:0>.
                      ! DEO3 Low Word
DILWORD
         = DE13<15:6>,
                      ! DE13 Low Word
D2LWORD
                      ! II[2] Low Word
         = DC23<15:0>,
                      ! DE31 Low Word
D3LWORD
         = D(3)<15:0>
DALWORD
                      ! D[4] Low Word
         = D[4]<15:0>,
                      ! DC5) Low Word
DSLWORD
         = DESI<15:0>,
                      ! I/[6] Low Word
DI LI WORD
         = DE63<15:0>,
D7LWORD
                      1 DE73 Low Word
         = DE73<15:0>,
DISKEGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
         = HANADR<15:0>, ! HANADR Low Word
HANADIRLOW
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI
         = TEMPADR<31:16>;! TEMPADR High Word
DIEPOTY
/#
                                                         1/
/*
                16K 16-Bit Word Internal Memory
                                                         */
/*
                                                         */
MEO:327673<7:0>;
sidero
/¥
/*
                Logic Level Macros
                                                         */
```

```
/*
10
    = 0 1,
    = 1 %,
hi
    = 0 1,
off
on
    = 1 %,
clear = 0 %;
/*
                                                       */
/* Fower On and Initialization. This process was not modeled but is
                                                       */
  added to initialize signals and registers.
                                                       */
                                                       */
power_on_initialize :=
      SWITCH = on;
                                ! Turn Power On
      next;
                                ! Execute Assignment
      READY = lo;
                                ! System Not Ready
      RESET = lo;
                                ! Assert Reset For
      delay(100);
                                ! 100 Miliseconds(Active Low)
      RESET = hi:
                                ! Deactivate Reset
                                ! Execute Pending Assignments
      next;
      ASN = hi;
                                ! Initialize Address Strobe
      LISN = hi:
                                ! Initialize Lower Data Strobe
      UDSN = hi;
                                ! Initialize Upper Data Strobe
      INTACKN = hi;
                                ! Initialize Data Transfer Acknowledge
      RW = hi;
                                ! Initialize Read/Write(Read On High)
      DBUS = 0xffff;
                               ! Flace Data Bus In High Impedance State
                                ! Place Memory Locations Following The
      M[0x100e] = 0xff;
      M[0x100f] = 0xff;
                                 ! JMP Instruction In A High State
      HALT = hi;
                                ! Initialize Halt Flip-Flop(Active
                                ! Low)
                                ! Initialize Clock Cycle Counter
      T = 0;
      READY = hi:
                                ! System Ready
      /*
                                                       */
      /*
           Routine Initialization Per Hamby and Guillory
                                                       */
      /*
                                                       */
      ! Place Hex 1004 Into ACOJ
      A[0] = 0 \times 1004;
      PC = 0x1000;
                                ! Place Hex 1000 Into Program Counter
      next
                                ! Execute Assignments
/*
                                                       */
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                       */
   to retrieve modeled instructions for simulation and analysis. It
                                                       */
   was fashsioned from the Read Cycle described by Hamby and Guillory */
/*
```

on page VI-15 of their thesis.

```
fetch_initial_instruction :=
     PHI1 = hi;
                                    ! Phase 1 Of
                                    ! Clock Cycle 0
    FHI2 = 10;
                                    ! Memory Read
    RW = hi;
                                    ! Disable Address Bus Buffer
    ADENABLE = 10;
                                    ! Disable Data Bus Buffer
    DRENABLE = 10:
    IABUS = PC;
                                    ! Place PC On Internal Address
    next;
                                    ! Execute Pending Assignments
                                    ! Phase 2 Of
    PHI1 = lo;
    PHI2 = hi;
                                    ! Clock Cycle 0
                                    ! Enable Address Bus Buffer
    ADENABLE = h1;
    EXABUF = IABUS;
                                    ! Gate Internal Address Bus
                                    ! Into External Address Ruffer
    FCMODE = SRMODE;
                                    ! User mode
    FCSFACE = 2;
                                    ! Accessing Program
                                    ! Execute Impending Assignments
    next;
    ABUS = EXABUF;
                                    ! Address Placed On Bus(Added)
    next;
                                    ! Execute Pending Assignments
     T = 1;
                                    ! Clock Cycle 1
    next;
                                    ! Execute Assignment
                                    ! Phase 1 Of
    PHI1 = hi;
                                    ! Clock Cycle 1
    PHI2 = 10:
     ASN = lo;
                                    ! Assert Address Strobe
                                    ! Assert Lower Data Strobe
    LISN = lo;
                                    ! Assert Upper Data Strobe
     UDSN = lo:
     DBENABLE = hi;
                                    ! Enable Data Rus
                                    ! Execute Pending Assignments
     next;
     PHI1 = lo;
                                    ! Phase 2
    PHI2 = hi;
                                    ! Of Clock Cycle 1
                                    ! Execute Fending Assignments
    next:
     T = 2;
                                    ! Clock Cycle 2
    next;
                                    ! Execute Assignment
    PHI1 = hi;
                                    ! Phase 1
    PHI2 = 10;
                                    ! Of Clock Cycle 2
    while DTACKN eql hi
                                    ! Wait For Memory To Place
                                    ! Data On The Bus
```

next;

! Execute Impending Assignments

```
PHI1 = 10;
                               ! Phase 2
    PHI2 = hi:
                               ! Of Clock Cycle 2
    next;
                               ! Execute Assignments
     T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
    next;
    PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 3
    FHI2 = 10;
     DBUS<15:8> = MEABUS3;
                               ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
                               ! Asserts DTACKN(Added)
    DITACKN = lo;
                               ! Execute Fending Assignments
    next;
     ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
                               ! Execute Impending Assignments
    next;
! Clock Cycle 3
T = 3;
next;
                               ! Execute Assignment
                               ! Phase 2
PHI1 = 1c;
                               ! Of Clock Cycle 3
PHI2 = hi;
                               ! Instruction On Bata Bus
EXDBUF = DBUS;
                               ! Is Placed In External Data
                               ! Bus Buffer
next;
                               ! Execute Pending Assignments
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase &
FHI2 = 10:
                                ! Of Clock Cycle 4
FFR = EXDBUF;
                                ! The Contents Of The External
                                ! Data Bus Ruffer Are Placed
                                ! In Prefetch Register
                                ! Execute Pending Assignments
next;
PHI1 = 10;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 4
ASN = hi;
                                ! Deactivate Address Strobe
                                ! Deactivate Lower Data Strobe
LDSN = hi;
                                ! Deactivate Upper Data Strobe
UDSN = hi;
                                ! Contents Of Prefetch Register
IR = PFR;
                                ! Are Placed Into Instruction
                                ! Keaister
                                ! Deactivate Data Transfer(Added)
DTACKN = hi;
```

```
! Acknowledge
                                        ! Increment Program Counter
     PC = PC + 4;
                                        ! Execute Pending Assignments
     next:
                                        ! Reset Clock Cycle Counter
     T = 0
                                        ! AND.W #$DFFF,SR
andi :=
    SRMODE = 10;
                                        ! Effect Of Instruction
    IR<15:8> = MEPCJ;
                                        ! Prefetch Next Instruction
    IR<7:0> = MEPC + 13;
                                        ! Is To Set Status Register
      PC = PC + 2;
                                          ! Increment Program Counter
    T = 5;
                                        ! Supervisor Bit To User
    next;
                                        ! Mode
    T = 0
                                        ! Requires 6 Clock Cycles
move :=
                                        ! MOVE.W #$555, [1]
     PHI1 = hi;
                                        ! Phase 1 Of
                                        ! Clock Cycle 0
     FHI2 = 10;
     DBUS = Oxffff;
                                        ! Place Data Bus In High Impedance
     RW = hi;
                                        ! Memory Read
     ADENABLE = lo;
                                       ! Disable Address Bus Buffer
     ABUS = 0xffffff;
                                       ! Address Bus High Impedanced
                                       ! Disable Data Bus Buffer
     DBENABLE = 10;
                                       ! Place PC On Internal Address
     labus<31:1> = PC<31:1>;
                                        ! Rus
                                        ! Execute Fending Assignments
     next;
                                       ! Phase 2 Of
     PHI1 = lo:
                                       ! Clock Cycle O
     PHI2 = hi:
     ABENABLE = hi;
                                       ! Enable Address Bus Buffer
                                       ! Gate Internal Address Bus
     EXABUF = IABUS<23:1>;
                                       ! Into External Address Buffer
     FCMODE = SRMODE;
                                       ! User Mode
     FCSPACE = 2;
                                        ! Accessing Program
     SRCARRY = 10;
                                       ! Clear Status Register Carry Bit
                                       ! Clear Status Register Overflow Bit
     SKOVER = lo;
     SRZERO = 10;
                                       ! Clear Status Register Zero Bit
     SENEG = 10;
                                       ! Clear Status Register Negative Bit
     ABUS = IABUS<23:1>;
                                       ! Place PC On Address Bus (Added)
     next:
                                        ! Execute impending Assignments
     ! Clock Cycle 1
     T = 1;
     next;
                                        ! Execute Assignment
     PHI1 = hi;
                                       ! Phase 1 Of
```

```
! Clock Cycle 1
PHI2 = 10;
                               ! Assert Address Strobe
ASN = 10;
                              ! Assert Lower Data Strobe
LISN = 10;
UDSN = lo;
                              ! Assert Upper Data Strobe
                              ! Enable Data Bus
DBENABLE = hi;
next;
                               ! Execute Pending Assignments
                               ! Phase 2
PHI1 = 10;
PHI2 = hi;
                              ! Of Clock Cycle 1
next;
                              ! Execute Pending Assignments
! Clock Cycle 2
                               ! Execute Assignment
next;
PHI1 = hi;
                              ! Phase 1
                              ! Of Clock Cycle 2
FHI2 = 10;
while DTACKN eql hi
                              ! Wait For Memory To Flace
                              ! Data On The Bus
    next;
                              ! Execute Impending Assignments
                              ! Phase 2
     FHI1 = lo;
                              ! Of Clock Cycle 2
    PHI2 = hi:
                              ! Execute Assignments
    next;
    T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
    next;
                               ! Phase 1
    FHI1 = hi;
    PHI2 = 10;
                               ! Of Clock Cycle 3
    INBUS<15:8> = MEABUSJ;
                               ! Memory Places Instruction
     DBUS<7:0> = MCABUS + 13;
                              ! On Data Bus And
    ITACKN = 10;
                               ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
    next:
     ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
                               ! Execute Impending Assignments
     next;
! Clock Cycle 3
T = 3:
next;
                               ! Execute Assignment
PHI1 = 10;
                              ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
nexti
```

```
T = 4;
                                 ! Clock Cycle 4
                                 ! Execute Assignment
next:
                                 ! Phase 1
PHI1 = hi;
                                 ! Of Clock Cycle 4
PHI2 = 10:
IDBUS = EXDBUF;
                                 ! Set Status Register
if EXDBUF eql 0
  SRZERO = hi;
                                 ! Bits As Appropriate
if EXDBUF<15> eql 1
  SRNEG = hi;
                                 ! Execute Pending Assignments
next;
PHI1 = 10;
                                 ! Phase 2
PHI2 = hi:
                                 ! Of Clock Cycle 4
                                 ! Deactivate Address Strobe
ASN = hi;
                                 ! Deactivate Lower Data Strobe
LIISN = hi;
                                 ! Deactivate Upper Data Strobe
UDSN = hi;
PC = PC + 2;
                                 ! Increment PC
DTACKN = hi;
                                 ! Deactivate Data Transfer(Added)
                                 ! Acknowledge
D[1] = IDBUS;
                                 ! Place Contents Of Internal
                                 ! Data Bus Into D[2]
                                 ! Execute Impending Assignments
next:
! Clock Cycle 5
T = 5:
next;
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 5
                                 ! Place Data Bus In High Impedance
IIRUS = 0xffff;
                                 ! Memory Read
RW = hi;
                                 ! Disable Address Bus Buffer
ADENABLE = 10;
ABUS = 0xffffff;
                                 ! Address Bus High Impedanced
DRENABLE = 10;
                                 ! Disable Data Bus Buffer
                                 ! Place PC On Internal Address
IABUS<31:1> = PC<31:1>;
next;
                                 ! Execute Pending Assignments
                                 ! Phase 2 Of
PHI1 = 10;
PHI2 = hi;
                                 ! Clock Cycle 5
                                ! Enable Address Bus Buffer
ADENABLE = hi;
EXABUF = IABUS<23:1>;
                                ! Gate Internal Address Bus
                                 ! Into External Address Buffer
                                 ! User Mode
FCMODE = SRMODE:
                                 ! Accessing Program
FCSPACE = 2;
ABUS = IABUS<23:1>;
                                ! Place PC On Address Bus
                                 ! Execute Impending Assignments
next;
T = 6;
                                 ! Clock Cycle 6
```

```
next;
                               ! Execute Assignment
                               ! Phase 1 Of
PHI1 = hi;
                               ! Clock Cycle 6
PHI2 = 10;
ASN = 10;
                               ! Assert Address Strobe
LDSN = 10;
                               ! Assert Lower Nata Strobe
UDSN = 10;
                               ! Assert Upper Data Strobe
                               ! Enable Data Bus
DBENABLE = hi;
next;
                               ! Execute Fending Assignments
PHI1 = lo:
                               ! Phase 2
PHI2 = hi:
                               ! Of Clock Cycle 6
next;
                               ! Execute Pending Assignments
T = 7;
                               ! Clock Cycle 7
next;
                               ! Execute Assignment
PHI1 = hi:
                              ! Phase 1
                              ! Of Clock Cycle 7
PHI2 = 10:
                              ! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
    next;
                               ! Execute Impending Assignments
    PHI1 = 1c:
                              ! Phase 2
    PHI2 = hi;
                              ! Of Clock Cycle 7
                              ! Execute Assignments
    next;
    ! Clock Cycle 8
                              ! Execute Assignment
    next;
    PHI1 = hi;
                              ! Phase 1
    PHI2 = 10;
                              ! Of Clock Cycle 8
    DBUS<15:8> = MEABUS];
                              ! Memory Places Instruction
    DBUS<7:0> = NEABUS + 13;
                              ! On Data Bus And
                               ! Asserts DTACKN(Added)
    DITACKN = 10;
    next;
                               ! Execute Pending Assignments
    ! Return To Phase 2
    T = 7
                               ! Of Clock Cycle 7
    );
    next;
                              ! Execute Impending Assignments
T = 8:
                               ! Clock Cycle 8
next;
                               ! Execute Assignment
PHI1 = lo;
                               ! Phase 2
PHI2 = hi:
                              ! Of Clock Cycle 8
EXDRUF = DBUS;
                              ! Instruction On Data Bus
                               ! Is flaced In External Data
```

```
! Bus Buffer
     next;
                                         ! Execute Pending Assignments
     T = 9;
                                         ! Clock Cycle 9
     next;
                                         ! Execute Assignment
     PHI1 = hi;
                                         ! Phase 1
     PHI2 = 1o;
                                         ! Of Clock Cycle 9
     PFR = EXDBUF;
                                         ! The Contents Of The External
                                         ! Data Bus Buffer Are Placed
                                         ! In Frefetch Register
     next;
                                         ! Execute Pending Assignments
     PHI1 = 10;
                                         ! Phase 2
     PHI2 = hi:
                                         ! Of Clock Cycle 9
     ASN = hi:
                                         ! Deactivate Address Strobe
     LDSN = hi;
                                         ! Neactivate Lower Nata Strobe
     UDSN = hi;
                                         ! Deactivate Upper Data Strobe
     PC = PC + 2;
                                         ! Increment Program Counter
                                         ! Are Placed Into Instruction
                                         ! Register
     DTACKN = hi:
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
     IR = PFR:
                                         ! Load Instruction Register With
                                         ! Frefetch Register
     next;
     T = 0
                                        ! JMF (A0)
JMP :=
     PHI1 = hi;
                                         ! Phase 1 Of
     FHI2 = 10;
                                         ! Clock Cycle 0
     DBUS = 0xffff;
                                        ! Place Data Bus In A High Impedance
     F_iW = hi;
                                         ! Memory Read
     ADENABLE = 10:
                                        ! Disable Address Bus Buffer
     DRENARLE = 10;
                                        ! Disable Nata Bus Buffer
     IABUS = PC;
                                        ! Place PC On Internal Address
     next;
                                        ! Execute Pending Assignments
     FHI1 = 10;
                                        ! Phase 2 Of
     PHI2 = hi;
                                        ! Clock Cycle 0
     ALIENABLE = hi;
                                        ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                        ! Gate Internal Address Bus
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSPACE = 2;
                                       .! Accessing Program
```

```
! Execute Pending Assignments
next;
ABUS = EXABUF;
                               ! Address Flaced On Rus(Added)
                               ! Execute Pending Assignments
next;
T = 1;
                               ! Clock Cycle 1
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1 Of
PHI2 = 10;
                               ! Clock Cycle 1
                               ! Assert Address Strobe
ASN = lo;
LDSN = lo;
                               ! Assert Lower Data Strobe
UDSN = lo;
                               ! Assert Upper Data Strobe
IABUS = A[0];
                               ! Move Jump Address From A[0]
                               ! To Internal Address Ruffer
                               ! Enable Data Rus
DBENABLE = hi;
                               ! Execute Pending Assignments
next;
PHI1 = lo;
                               ! Phase 2
PHI2 = ni;
                               ! Of Clock Cycle 1
PC = IABUS;
                               ! Flace Jump Address Into Program
                               ! Counter
next:
T = 2;
                               ! Clock Cycle 2
next:
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
FHI2 = 10;
                               ! Wait For Memory To Place
while DTACKN eql hi
                               ! Nata On The Rus
     next;
                               ! Execute Impending Assignments
     PHI1 = 10:
                               ! Phase 2
     PHI2 = hi;
                               ! Of Clock Cycle 2
     next;
                               ! Execute Assignments
     T = 3;
                               ! Clock Cycle 3
                               ! Execute Assignment
     next;
                               ! Phase 1
     PHI1 = hi;
                               ! Of Clock Cycle 3
     PHI2 = 16:
     INBUS<15:8> = MEABUSJ;
                               ! Memory Places Instruction
                               ! On Data Bus And
     DBUS<7:0> = MEABUS + 13;
     TITACKN = 10;
                               ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
     next;
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
```

```
! Execute Impending Assignments
     next;
! Clock Cycle 3
7 = 3;
next;
                                 ! Execute Assignment
PHI1 = 10;
                                 ! Phase 2
                                 ! Of Clock Cycle 3
PHI2 = hi;
                                 ! Instruction On Data Bus
EXDEUF = DRUS;
                                 ! Is Placed In External Data
                                 ! Bus Buffer
next:
                                 ! Execute Pending Assignments
! Clock Cycle 4
next;
                                 ! Execute Assignment
PHI1 = hi:
                                 ! Phase 1
                                 ! Of Clock Cycle 4
PHI2 = 10;
next;
FFR = EXDBUF;
                                 ! The Contents Of The External
                                 ! Nata Bus Buffer Are Placed
                                 ! In Prefetch Register
                                 ! Execute Pending Assignments
next:
FHI1 = 10;
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 4
ASN = hi;
                                 ! Deactivate Address Strobe
LOSN = hi;
                                 ! Deactivate Lower Data Strobe
                                 ! Deactivate Upper Data Strobe
UDSN = hi;
                                 ! Deactivate Data Transfer
DTACKN = hi;
                                 ! Acknowledge(Added)
next;
T = 5;
                                 ! Clock Cycle 5
next;
                                 ! Execute Previous Assignment
FHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 5
RW = hi;
                                 ! Memory Read
AMENABLE = lo;
                                 ! Disable Address Bus Buffer
DBENABLE = 10;
                                 ! Disable Data Bus Buffer
TABUS = PC:
                                 ! Place PC On Internal Address
                                 ! Bus
next:
                                 ! Execute Pending Assignments
                                 ! Phase 2 Of
PHI1 = 10;
PHI2 = hi;
                                 ! Clock Cycle 5
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSFACE = 2;
                                 ! Accessing Program
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
next;
                                 ! Into External Address Buffer
```

```
! Address Placed On Bus(Added)
ABUS = EXABUF;
                              ! Execute Pending Assignments
next;
! Clock Cycle 6
T = 6;
next;
                              ! Execute Assignment
FHII = hi;
                              ! Phase 1 Of
PHI2 = 10;
                              ! Clock Cycle 6
ASN = 10;
                              ! Assert Address Strobe
LUSN = 10;
                              ! Assert Lower Data Strobe
UI(SN = 10;
                              ! Assert Upper Data Strobe
DRENABLE = hi;
                              ! Enable Data Bus
next;
                              ! Execute Pending Assignments
FHI1 = 10;
                              ! Phase 2
                              ! Of Clock Cycle 6
PHI2 ≈ hi;
next;
                              ! Execute Pending Assignments
! Clock Cycle 7
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1
                             ! Of Clock Cycle 7
PHI2 = 10;
while DTACKN eql hi
                             ! Wait For Memory To Place
                              ! Data On The Rus
                              ! Execute Impending Assignments
    next;
    fHI1 = lo;
                              ! Phase 2
    PHI2 = hi;
                              ! Of Clock Cycle 7
                              ! Execute Assignments
    next;
    T = 8;
                             ! Clock Cycle 8
    next;
                             ! Execute Assignment
    PHI1 = hi;
                              ! Phase 1
    PHI2 = 10;
                              ! Of Clock Cycle 8
    DBUS <15:8> = MEABUS3;
                              ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                             ! On Data Bus And
    INTACKN = 10:
                              ! Asserts DTACKN(Added)
                              ! Execute Pending Assignments
    next;
     T = 7
                             ! Return To Phase 2
                             ! Of Clock Cycle 7
    );
    next;
                             ! Execute Impending Assignments
T = 8;
                              ! Clock Cycle 8
next;
                              ! Execute Assignment
```

```
! Phase 2
     PHI1 = lo;
                                          ! Of Clock Cycle 8
     PHI2 = hi;
     EXDBUF = DBUS;
                                          ! Instruction On Nata Bus
                                          ! Is Placed In External Data
                                          ! Rus Ruffer
                                          ! Execute Fending Assignments
     next;
     T = 9;
                                          ! Clock Cycle 9
                                          ! Execute Assignment
     next;
                                          ! Phase 1
     PHI1 = hi;
                                          ! Of Clock Cycle 9
     PHI2 = 10;
                                          ! The Contents Of The External
     PFR = EXDRUF;
                                          ! Data Bus Buffer Are Placed
                                          ! In Prefetch Register
     next;
                                          ! Execute Pending Assignments
     PHI1 = lo;
                                          ! Phase 2
                                          ! Of Clock Cycle 9
     PHI2 = hi;
                                          ! Deactivate Address Strobe
     ASN = hi;
                                          ! Deactivate Lower Data Strobe
     LDSN = hi;
                                          ! Deactivate Upper Data Strobe
     UDSN = hi;
                                          ! Increment Program Counter
     PC = PC + 2;
                                          ! Place Contents Of Prefetch
     IR = PFR;
                                          ! Register Into Instruction
                                          ! Register
                                          ! Deactivate Data Trunsfer
     DTACKN = hi;
                                          ! Acknowledge(Added)
                                          ! Execute Pending Assignments
     next;
                                          ! Reset Clock Cycle Counter
     T = 0
     )
decode_execute_prefetch :=
                       case IR
                           0x323c: move ! MOVE.W #$5555,D1
                           0x027c: andi
                                          ! AND.W #$DFFF,SR
                           047320: jmp
                                          ! JMP (AO) If IR = Octal Value
                       esac
                       )
main :=
    power_on_initialize;
    fetch_initial_instruction;
    while READY eql hi
          decode_execute_prefetch
    )
```

```
/x
                                                  1/
/*
    MOTOROLA MC68000 MODEL OF THE ADD.W 03,05 INSTRUCTION
                                                  */
/¥
/*
/*
              Structure Declarations
                                                  */
/x
                                                  */
state
/*
                                                  */
/*
            M68000 Programming Registers
                                                  */
/*
                                                  */
DE0:73<31:0>,
                  🦈 ! 8 Nata Registers
AE0:63<31:0>,
                   ! 7 Address Registers
                   ! User Stack Pointer
UA7<31:0>,
                     System Stack Pointer
SA7<31:0>,
PC<31:0>,
                   ! Program Counter
SR<15:0>.
                   ! Status Register
/*
                                                  */
                                                  1/
/*
            Temporary Internal Registers
/*
                                                  x/
PFR<15:0>,
                   ! Prefetch Register
IR<15:0>,
                   ! Instruction Register
FC<2:0>,
                   ! Function Code Register
EXDRUF<15:0>,
                   ! External Data Bus Buffer Register
EXABUF<23:1>.
                   ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,
                   ! ALU Ruffer 1
                   ! ALU Ruffer 2
ALUBUF2<31:0>,
                   ! Temporary Data Storage
DIEMP<15:0>.
                   ! Temporary Displacement Storage
I(ISREG<31:0>,
SRTEMP<15:0>,
                   ! lemporary Status Register Storage
                    ! (Exception Processing)
IRTEMP<15:0>,
                     Temporary Instruction Register Storage
                    ! (Exception Processing)
                   ! Temporary Cycle Address Storage
TEMPAUR<31:0>,
                    ! (Exception Processing)
ACTYPE<15:0>,
                    ! Temporary Access Type Storage
                    ! (Exception Processing)
VECADR<2310>,
                   ! Temporary Vector Address Storage
                    ! (Exception Processing)
```

```
HANADR<31:0>.
                         ! Temporary Address Storage For
                         ! Exception Handler Routine
T<7:0>,
                         ! Clock Cycle Counter
                      ! Reset Flip-Flop
RESET,
HALT,
                      ! Halt Flip-Flop
                      ! Read/Write Flip-Flop
RW.
ADENABLE,
                      ! Address Bus Buffer Enable
DBENABLE,
                      ! Nata bus Buffer Enable
ASN,
                      ! Address Strobe Flip-Flop
LIISN,
                      ! Lower Nata Strobe Flip-Flop
UDSN,
                      ! Upper Data Strobe Flip-Flop
DITACKN.
                      ! Nata Transfer Acknowledge Flip-Flop
COUT,
                      ! Carry Flip-Flop
EXCEPT,
                      ! Exception Processing Flip-Flop
READY,
                      ! Ready Flip-Flop
/ x
                                                               */
/*
       Model transformation modifications:
                                                               */
/*
                                                               */
/#
           1) CDL decoder structure nonexistent in ISP' and un-
                                                               */
/*
       necessary for model. Eliminated.
                                                               */
/x
           2) Multi-phase clock structure nonexistent in ISP'.
                                                               x/
/*
       Operations on registers will provide its equivalent.
                                                               x/
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                               ¥/
                                                               */
       register_will provide its equivalent.
/*
1*

    The declared bus structures are modeled with registers */

/*
       without loss of model accurracy. This done to maintain model
                                                               */
                                                               */
/*
       equivalency and simplicity.
/*
           5) The memory word length was reduced from 16 to 8 bit
                                                               */
       words to coincide with the ECH's 32-Kbyte memory, to agree with*/
/*
/*
       their PC incrementation, and to enable the use of existing
                                                               */
/*
       MC68000 assembler and linker/loader models. The memory was
                                                               */
       also reduced from 8 Mwords to 32 Kbytes.
                                                               */
/1
                                                               */
/ x
IABUS<31:0>,
                         ! Internal Address Bus
IDBUS<31:0>,
                         ! Internal Data Bus
SWITCH,
                      ! Power Switch
PHI1.
                      ! Phase 1 Of Two-Phase Clock
                      ! Phase 2 Of Two-Phase Clock
PHI2;
port
*/
/x
/*
                                                               */
              External Address and Data Rus
/*
                                                               */
DBUS<15:0>.
                        ! External Data Bus
ABUS<23:1>;
                        ! External Address Bus(changed)
```

```
format
/*
/*
                                                         */
                Register Subfields
/#
                                                         */
PCADDR
         = PC<23:0>,
                      ! Frogram Counter Address Field
SRTRACE
         = SR<15>,
                      ! Trace Rit
SKHODE
                        Mode Selection Bit
         = SR<13>,
         = SR<0>,
                      ! Carry Est
SRCARRY
         = SR<1>,
SKOVER
                      ! Overflow Bit
         = SR<2>,
SRZERO
                      ! Zero Bit
SRNEG
         = SR<3>.
                      ! Negative Bit
SREX
         = SR<4>,
                      ! Extend Bit
SRMASK
         = Sk<10:8>,
                      ! Interrupt Mask
FCSPACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCHODE
         = FC<2>,
                      ! User/Supervisor Mode Bit
FCLOW
         = PC<15:0>,
                      ! PC Low Word
PCHI
                      ! PC High Word
         = PC<31:16>,
DOLWORD
                      ! II[0] Low Word
         = DE03<15:0>,
DILWORD
         = bc13<15:0>.
                      ! DC13 Low Word
D2LWORD
                      ! B[2] Low Word
         = 10[2]\langle 15:0 \rangle,
                      ! DE33 Low Word
D3LWORD
         = D(3)(15:0),
I/4LWORD
         = D[4]<15:0>.
                      ! NT43 Low Word
DSLWORD
         = BE53<15:0>,
                      ! D(5) Low Word
DIGLWORD
         = DE63<15:0>,
                      ! D[6] Low Word
DZLWORD
         = DE73<15:0>,
                      ! DE73 Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW
         = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADRLOW
         = TEMPADR<15:0>,! TEMPADR Low Word
         = TEMPADR<31:16>;! TEMPADR High Word
TEMPADRHI
DEMOTY
/*
                                                         */
                                                         */
/*
                 16K 16-Bit Word Internal Memory
                                                         1/
/*
ME0:327673<7:0>:
BIGCTO
/*
                                                         */
                                                         1/
/*
                Logic Level Macros
                                                         1/
/x
```

```
l۸
     = 0 1,
     = 1 2,
hi
off
    = 0 1.
CID
    = 1 %,
clear = 0 1;
*/
/* Power On and Initialization. This process was not modeled but is
                                                          1/
                                                          */
/* added to initialize signals and registers.
                                                          */
power_on_initialize :=
      SWITCH = on;
                                  ! Turn Power On
      next:
                                  ! Execute Assignment
      READY = lo;
                                  ! System Not Ready
      RESET = 10;
                                  ! Assert Reset For
      delay(100);
                                  ! 100 Miliseconds(Active Low)
                                  ! Deactivate Reset
      RESET = hi;
      next:
                                 ! Execute Pending Assignments
                                  ! Initialize Address Strobe
      ASN = hi;
                                  ! Initialize Lower Data Strobe
      LDSN = hi;
      UDSN = hi:
                                 ! Initialize Upper Data Strobe
                                 ! Initialize Data Transfer Acknowledge
      DTACKN = hi;
                                 ! Initialize Read/Write(Read On High)
      RW = hi;
      INUS = Oxffff;
                                 ! Place Nata Bus In High Impedance State
      MEO \times 10083 = O \times ff;
                                 ! Place Memory Locations Following The
      M[0x1009] = 0xff;
                                  ! JMP Instruction In A High State
      HALT = hi;
                                  ! Initialize Halt Flip-Flop(Active
                                  ! Low)
      T = 0:
                                  ! Initialize Clock Cycle Counter
      READY = hi;
                                  ! System Ready
      /x
                                                          */
                                                          */
      /*
            Routine Initialization Per Hamby and Guillory
      /*
                                                          */
      ! Set Status Register To User Mode
      SRMODE = 10;
      I(1] = 0.00000003;
                                  ! Place Hex 00000003 Into D[1]
                                  ! Place Hex 00000002 Into BE2]
      I(3) = 0.00000002;
      D[5] = 0x000000000;
                                 ! Initialize D[5] To Zero
      A[0] = 0 \times 1000;
                                  ! Place Hex 1000 Into A[0]
                                 ! Store Data At Hex 2000
      A[2] = 0 \times 2000;
      FC = 0 \times 1000;
                                 ! Place Hex 1000 Into Program Counter
      next
                                 ! Execute Assignments
      )
```

/*

```
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
/# to retrieve modeled instructions for simulation and analysis. It
                                                            */
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/* on page VI-15 of their thesis.
                                                            */
/*
                                                            */
fetch_initial_instruction !=
     ! Phase 1 Of
     PHI1 = hi;
                                     ! Clock Cycle 0
     FHI2 = 10:
                                     ! Memory Read
     RW = hi;
                                     ! Disable Address Rus Buffer
     ABENABLE = 10;
                                     ! Disable Data Bus Buffer
     DBENABLE = 10;
     IABUS = PC;
                                     ! Place PC On Internal Address
                                     ! Bus
     next;
                                     ! Execute Pending Assignments
                                     ! Phase 2 Of
     PHI1 = lo;
     PHI2 = hi;
                                     ! Clock Cycle O
                                     ! Enable Address Bus Buffer
     ARENABLE = hi;
                                     ! Gate Internal Address Bus
     EXABUF = IABUS;
                                     ! Into External Address Buffer
                                     ! User Mode
     FCMODE = SRMODE;
     FCSPACE = 2:
                                     ! Accessing Program
                                     ! Execute Impending Assignments
     next;
     ARUS = EXABUF;
                                     ! Address Flaced On Rus(Added)
     next:
                                     ! Execute Pending Assignments
     T = 1;
                                     ! Clock Cycle 1
                                     ! Execute Assignment
     next;
                                     ! Phase 1 Of
     PHI1 = hi;
                                     ! Clock Cycle 1
     PHI2 = 10;
                                     ! Assert Address Strobe
     ASN = 10;
                                     ! Assert Lower Data Strobe
     LUSN = lo;
     UIISN = 10:
                                     ! Assert Upper Data Strobe
     DBENABLE = hi;
                                     ! Enable Data Rus
                                     ! Execute Pending Assignments
     next;
                                     ! Phase 2
     PHI1 = lo;
                                     ! Of Clock Cycle 1
     PHI2 = hi;
     next;
                                     ! Execute Pending Assignments
     ! Clock Cycle 2
     T = 2;
                                     ! Execute Assignment
     next;
                                     ! Phase 1
     FHII = hi;
```

```
PHI2 = 10:
                               ! Of Clock Cycle 2
while DTACKN eql hi
                               ! Wait For Memory To Place
                               ! Data On The Bus
                               ! Execute Impending Assignments
    next;
    PHI1 = lo;
                               ! Phase 2
    PHI2 = hi;
                               ! Of Clock Cycle 2
    next;
                               ! Execute Assignments
    T = 3;
                               ! Clock Cycle 3
    next:
                               ! Execute Assignment
                               ! Phase 1
    PHI1 = hi;
    PHI2 = 10;
                               ! Of Clock Cycle 3
    DBUS<15:8> = MCABUSJ;
                               ! Memory Places Instruction
    DBUS<7:0> = m[ABUS + 1];
                               ! On Data Rus And
                               ! Asserts DTACKN(Added)
     DTACKN = lo:
    next;
                               ! Execute Fending Assignments
    ! Return To Phase 2
                               ! Of Clock Cycle 2
    );
                              ! Execute Impending Assignments
    next;
T = 3:
                               ! Clock Cycle 3
next; .
                               ! Execute Assignment
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DBUS:
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PHI1 = hi:
                               ! Phase 1
PHI2 = 10:
                               ! Of Clock Cycle 4
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
FHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                               ! Deactivate Address Strobe
                               ! Deactivate Lower Data Strobe
LIISN = hi;
UDSN = hi;
                               ! Deactivate Upper Data Strobe
```

```
IR = PFR:
                                        ! Contents Of Prefetch Register
                                        ! Are Placed Into Instruction
                                        ! Register
     DTACKN = hi;
                                        ! Deactivate Nata Transfer(Added)
                                        ! Acknowledge
     PC = PC + 2;
                                        ! Increment Program Counter
     next;
                                        ! Execute Pending Assignments
     T = 0
                                        ! Reset Clock Cycle Counter
odd :=
                                        ! ADD.W D3,D5
     ! Phase 1 Of
     PHI1 = hi;
     FHI2 = 10;
                                        ! Clock Cycle 0
     DBUS = Oxffff;
                                        ! Place Data Bus In High Impedance
     RW = hi;
                                        ! Memory Read
     IDBUS = D3LWORD;
                                        ! Place Low Word Of D[3]
                                        ! Onto Internal Data Bus
     ADENABLE = 10;
                                        ! Disable Address Bus Buffer
     DRENABLE = 10;
                                        ! Disable Data Rus Buffer
     IABUS = PC;
                                        ! Place PC On Internal Address
     next;
                                        ! Execute Pending Assignments
     FHI1 = lo;
                                        ! Phase 2 Of
     PHI2 = hi;
                                        ! Clock Cycle 0
     ADENABLE = hi;
                                        ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                        ! Gate Internal Address Bus
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSPACE = 2;
                                        ! Accessing Program
     ALUBUF1 = IDBUS;
                                        ! Place Data From Internal Data Bus
                                        ! Into ALU Buffer 1
                                        ! Execute Impending Assignments
     next;
     ABUS = EXABUF;
                                        ! Address Placed On Bus(Added)
                                        ! Execute Pending Assignments
     next;
     T = 1:
                                        ! Clock Cycle 1
     next;
                                        ! Execute Assignment
     PHI1 = hi;
                                        ! Phase 1 Of
     PHI2 = 10;
                                        ! Clock Cycle 1
      3N = 10:
                                        ! Assert Address Strobe
     LISN = lo:
                                        ! Assert Lower Data Strobe
                                        ! Assert Upper Data Strobe
     UDSN = 10;
     DBENABLE = hi;
                                        ! Enable Data Bus
                                        ! Place Low Word From B[5]
     IDBUS = D5LWORD:
                                        ! Onto Internal Data Bus
```

! Execute Pending Assignments

next;

```
PHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 1
                               ! Flace Nata From Internal Nata
ALUBUF2 = IDBUS;
                               ! Bus Into ALU Buffer 2
                               ! Execute Pending Assignments
next;
T = 2;
                               ! Clock Cycle 2
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 2
IDBUS = ALUBUF1 + ALUBUF2;
                              ! Place Sum From ALU Onto
                              ! Internal Data Bus
                               ! Reset Condition Codes
SRCARRY = lo:
SROVER = 10;
SRZERO = lo;
SRNEG = 10;
SREX = 10;
                              ! Wait For Memory To Place
while DTACKN eql hi
                              ! Data On The Bus
    next;
                              ! Execute Impending Assignments
    PHI1 = 10;
                              ! Phase 2
    PHI2 = hi;
                              ! Of Clock Cycle 2
    next:
                              ! Execute Assignments
     T = 3:
                               ! Clock Cycle 3
    next;
                               ! Execute Assignment
    PHI1 = hi;
                              ! Phase 1
                               ! Of Clock Cycle 3
    PHI2 = 10:
                              ! Memory Places Instruction
     DBUS<15:8> = M[ABUS];
    DBUS<7:0> = MEABUS + 13;
                             ! On Data Bus And
    DTACKN = 10;
                               ! Asserts DTACKN(Added)
    next;
                               ! Execute Fending Assignments
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
    );
                               ! Execute Impending Assignments
    next:
1 = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
FHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 3
EXDRUF = DBUS;
                               ! Instruction On Data Bus
```

! Is Placed In External Data

```
! Sum On Internal Data Bus
     DSLWORD = IDBUS;
                                        ! Is Place Into Low Word Of D[5]
     next;
                                        ! Execute Pending Assignments
     T = 4;
                                        ! Clock Cycle 4
                                        ! Execute Assignment
     next;
                                        ! Phase 1
     PHI1 = hi;
                                        ! Of Clock Cycle 4
     FHI2 = 10;
                                        ! Set Condition Code
     if D5LWORD eq1 0
        SRZERO = hi;
                                       ! Rits As Appropriate
     if COUT eql 1
        SRCARRY = hi;
        SREX = hi
        );
     if DC53<15>
        SRNEG = hi;
     SROVER = ((not D[53<15>) and ALUBUF1<15> and ALUBUF2<15>)
             or (BE53<15> and (not ALUBUF1<15>) and (not ALUBUF2<15>));
                                        ! The Contents Of The External
     PFR = EXDRUF;
                                        ! Data Bus Buffer Are Placed
                                        ! In Prefetch Register
                                        ! Execute Pending Assignments
     next;
     PHI1 = lo;
                                        ! Phase 2
     PHI2 = hi;
                                        ! Of Clock Cycle 4
                                        ! Deactivate Address Strobe
     ASN = hii
                                        ! Deactivate Lower Data Strobe
     LDSN = hi;
     UDSN = hi;
                                        ! Deactivate Upper Data Strobe
                                        ! Contents Of Prefetch Register
     IR = PFR;
                                        ! Are flaced Into Instruction
                                        ! Register
     DITACKN = hi;
                                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
     PC = PC + 2;
                                        ! Increment Program Counter
                                        ! Execute Impending Assignments
     next;
                                        ! Reset Clock Cycle Counter
     T = 0
                                        ! MOVE.W D1,D5
moved :=
     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                        ! Clock Cycle 0
                                        ! Place Data Bus In High Impedance
     DBUS = 0xffff;
                                        ! Hemory Read
     RW = hi;
                                        ! Disable Address Bus Buffer
     ADENABLE = 10;
                                        ! Disable Data Bus Buffer
     DIBENABLE = 10;
```

! Bus Buffer

```
IABUS = PC;
                                   ! Place PC On Internal Address
                                   ! Bus
                                   ! Place Low Word From DC13 Onto
IDBUS = D1LWORD;
                                   ! Internal Data Bus
next;
                                   ! Execute Pending Assignments
                                   ! Phase 2 Of
PHI1 = 10;
PHI2 = hi;
                                   ! Clock Cycle O
ADENABLE = hi;
                                   ! Enable Address Rus Buffer
EXABUF = IABUS;
                                   ! Gate Internal Address Bus
                                   ! Into External Address Buffer
FCMODE = SRMODE;
                                   ! User Mode
FCSPACE = 2;
                                   ! Accessing Program
SRCARRY = lo;
                                   ! Clear Status Register Carry Bit
SROVER = 10;
                                   ! Clear Status Register Overflow Bit
                                   ! Clear Status Register Zero Bit
SRZERO = 10;
SRNEG = 10:
                                  ! Clear Status Register Negative Bit
                                   ! Place Data From Internal Data Bus
DSLWORD = IDBUS;
                                   ! Into Low Word Of B[2]
                                   ! Execute Impending Assignments
next;
ABUS = EXABUF;
                                   ! Address Placed On Bus(Added)
                                   ! Execute Pending Assignments
next;
T = 1;
                                   ! Clock Cycle 1
next;
                                   ! Execute Assignment
                                   ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                                   ! Clock Cycle 1
ASN = lo;
                                   ! Assert Address Strobe
LDSN = lo;
                                   ! Assert Lower Data Strobe
UDISN = lo;
                                   ! Assert Upper Data Strobe
DBENABLE = hi:
                                   ! Enable Bata Bus
                                   ! Set Status Register Zero Bit
if DSLWORD eql 0
   SRZERO = hi;
                                   ! If Moved Data Is Zero
next:
                                   ! Execute Pending Assignments
                                   ! Phase 2
fHI1 = 10;
                                   ! Of Clock Cycle 1
PHI2 = hi:
if DC53<15>
                                   ! Set Status Register Negative
  SKNEG = hi;
                                   ! kit If Moved Data Is Negative
                                   ! Execute Pending Assignments
T = 2;
                                   ! Clock Cycle 2
                                   ! Execute Assignment
next;
                                   ! Fhase 1
FHI1 = hi
                                   ! Of Clock Cycle 2
PHI2 = 10;
                                   ! Wait For Memory To Place
while DTACKN eql hi
                                   ! Data On The Bus
     next;
                                   ! Execute Impending Assignments
```

```
PHI1 = lo;
                               ! Phase 2
     PHI2 = hi;
                               ! Of Clock Cycle 2
                               ! Execute Assignments
     next;
     T = 3:
                               ! Clock Cycle 3
     next:
                               ! Execute Assignment
     PHI1 = hi;
                               ! Phase 1
     FHI2 = 10;
                               ! Of Clock Cycle 3
     DBUS<15:8> = MEABUS];
                               ! Memory Places Instruction
     IGBUS<7:0> = MEABUS + 13;
                               ! On Nata Bus And
     DITACKN = 10;
                               ! Asserts DTACKN(Added)
     next;
                               ! Execute Pending Assignments
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
     next;
                               ! Execute Impending Assignments
T = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
PHI1 = 10;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 3
EXDRUF = DBUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Ruffer
                               ! Execute Pending Assignments
next;
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 4
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
PHI1 = 10;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 4
ASN = hi:
                               ! Deactivate Address Strobe
LDSN = hi:
                               ! Deactivate Lower Data Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
IR = PFR;
                               ! Contents Of Prefetch Register
                                ! Are Placed Into Instruction
                                ! Register
DTACKN = hi;
                               ! Deactivate Nata Transfer(Added)
                               ! Acknowledge
```

```
PC = PC + 2;
                                     ! Increwent Program Counter
     next;
                                      ! Execute Impending Assignments
     T = 0
                                      ! Reset Clock Cycle Counter
                                       ! MOVE.W D5, (A2)
movei :=
     ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10:
                                      ! Clock Cycle 0
     DHUS = 0xffff;
                                      ! Place Nata Bus In High Impedance
     RW = hi;
                                      ! Memory Read
     ALIENABLE = 10;
                                      ! Disable Address Bus Buffer
     DBENABLE = 10;
                                      ! Disable Data Bus Buffer
     IABUS = PC;
                                      ! Place PC On Internal Address
                                      ! Kus
                                      ! Execute Fending Assignments
    next;
     FHI1 = 10;
                                      ! Phase 2 Of
                                      ! Clock Cycle O
     PHI2 = hi;
     ADENABLE = hi;
                                      ! Enable Address Bus Buffer
     EXABUF = IARUS;
                                     ! Gate Internal Address Bus
                                      ! Into External Address Buffer
     FCMODE = SRMODE;
                                      ! User Mode
                                      ! Accessing Program
     FCSPACE = 2;
                                      ! Execute Impending Assignments
     next;
     ABUS = EXABUF;
                                      ! Address Flaced On Rus(Added)
     next:
                                      ! Execute Pending Assignments
     T = 1:
                                      ! Clock Cycle 1
     next;
                                      ! Execute Assignment
     PHI1 = hi;
                                      ! Phase 1 Of
     PHI2 = 10;
                                      ! Clock Cycle 1
     ASN = lo;
                                      ! Assert Address Strobe
                                      ! Assert Lower Data Strobe
     LDSN = lo:
     UDSN = lo;
                                      ! Assert Upper Data Strobe
     DBENABLE = hi;
                                      ! Enable Data Bus
     next;
                                      ! Execute Pending Assignments
     FHI1 = lo;
                                      ! Phase 2
     PHI2 = hi;
                                      ! Of Clock Cycle 1
     next;
                                      ! Execute Fending Assignments
     T = 2;
                                      ! Clock Cycle 2
                                      ! Execute Assignment
     next;
     PHI1 = hi;
                                      ! Phase 1
                                      ! Of Clock Cycle 2
     PHI2 = 10;
```

```
while DTACKN eql hi
                               ! Wait For Memory To Place
                               ! Data On The Bus
                               ! Execute Impending Assignments
     next;
     PHI1 = 10:
                               ! Phase 2
     FHI2 = hi;
                                ! Of Clock Cycle 2
     next;
                                ! Execute Assignments
     ! Clock Cycle 3
     T = 3:
     next;
                                ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 3
     PHI2 = 10:
     DBUS<15:8> = MCABUS3;
                                ! Memory Places Instruction
     LBUS<7:0> = MEABUS + 13;
                                ! On Data Bus And
     DTACKN = lo;
                                ! Asserts DTACKN(Added)
     next:
                                ! Execute Pending Assignments
     ! Return To Phase 2
     T = 2
                                ! Of Clock Cycle 2
     );
     next;
                                ! Execute Impending Assignments
\Upsilon = 3;
                                ! Clock Cycle 3
next;
                                ! Execute Assignment
PHI1 = 10:
                                ! Phase 2
                                ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = LIBUS;
                               ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Bus Buffer
                                ! Execute Fending Assignments
next;
/*********************************
T = 4;
                               ! Clock Cycle 4
next;
                                ! Execute Assignment
PHI1 = bi;
                                ! Phase 1
PHI2 = 16:
                                ! Of Clock Cycle 4
                                ! The Contents Of The External
PFR = EXDBUF;
                                ! Data Bus Ruffer Are Placed
                                ! In Prefetch Register
                                ! Execute Fending Assignments
next;
PHI1 = 10
                                ! Phase 2
                                ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                                ! Deactivate Address Strobe
LUSN = hi;
                                ! Deactivate Lower Data Strobe
                                ! Deactivate Upper Data Strobe
UDSN = hi;
                                ! Are Flaced Into Instruction
```

```
! Register
DTACKN = hi;
                                   ! Neactivate Nata Transfer(Added)
                                   ! Acknowledge
next:
! Clock Cycle 5
T = 5;
next:
                                   ! Execute Previous Assignment
                                   ! Phase 1 Of
PHI1 = hi;
                                   ! Clock Cycle 5
PHI2 = 10;
                                   ! hemory Read
RW = hi;
                                   ! Disable Address Bus Buffer
ADENABLE = 10;
DBUS = 0 \times ffff:
                                   ! Return Data Bus To High
                                   ! Impedance State
                                   ! Disable Data Bus Buffer
DRENABLE = 10;
                                   ! flace A[2] On Internal Address
IABUS = A[2];
                                   ! Bus
next;
                                   ! Execute Pending Assignments
                                   ! Phase 2 Of
FHI1 = lo;
PHI2 = hi;
                                   ! Clock Cycle 5
                                   ! Enable Address Rus Buffer
ADENABLE = ha;
FCMODE = SRMODE;
                                   ! User Mode
FCSPACE = 1;
                                   ! Accessing Program
                                  ! Gate Internal Address Bus
EXABUF = IAHUS;
IDBUS = D5LWORD:
                                   ! Place Low Word from D[5] On
                                   ! Internal Data Bus
                                   ! Into External Address Buffer
next;
ABUS = EXABUF;
                                   ! Address Placed On Bus(Added)
                                   ! Execute Pending Assignments
next;
T = 6;
                                   ! Clock Cycle 6
                                  ! Execute Assignment
next;
PHI1 = hi;
                                   ! Phase 1 Of
                                   ! Clock Cycle 6
PHI2 = 10;
ASN = 10;
                                   ! Assert Address Strobe
RW = 10;
EXDBUF = IDBUS;
                                   ! Place Contents Of Internal
                                   ! Data Bus Into External Data Buffer
                                   ! Reset Condition Code Bits
SRCARRY = 10;
SROVER = 10;
SRZERO = 10;
SKNEG = 10:
next;
                                   ! Execute Fending Assignments
                                   ! Phase 2
PHI1 = lo:
                                   ! Of Clock Cycle &
PHI2 = hi;
                                   ! Set Zero Condition Bit If Needed
if EXDBUF eql 0
  SRZERU = hi;
                                   ! Place Data On External Data Bus
DBUS = EXDBUF;
```

```
DBENABLE = h1:
                             ! Enable Data Bus
next;
                             ! Execute Pending Assignments
! Clock Cycle 7
                             ! Execute Assignment
next;
PHI1 = hi;
                             ! Phase 1
PHI2 = 10;
                             ! Of Clock Cycle 7
if EXDBUF<15>
                             ! Set Negative Condition Bit
  SRNEG = hi;
                             ! If Needed
                             ! Activate Upper And
UIISN = lo;
                             ! Lower Data Strobes
LUSN = 10;
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Ilata On The Bus
    next;
                             ! Execute Impending Assignments
    PHI1 = lo;
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 7
    next;
                             ! Execute Assignments
    T = 8;
                             ! Clock Cycle 8
    next:
                             ! Execute Assignment
    PHI1 = hi;
                             ! Phase 1
    PHI2 = 10;
                             ! Of Clock Cycle 8
    mcabus] = DBUS<15:8>;
                             ! Store Data From Bus
    MEARUS + 1] = DRUS<7:0>;
                             ! In Memory
    DIACKN = 10;
                             ! Asserts DTACKN(Added)
                             ! Execute Fending Assignments
    next:
    T = 7
                             ! Return To Phase 2
                             ! Of Clock Cycle 7
    );
                            ! Execute Impending Assignments
    next;
T = 8;
                             ! Clock Cycle 8
next;
                             ! Execute Assignment
FHI1 = lo;
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 8
next;
                             ! Execute Fending Assignments
T = 9:
                              ! Clock Cycle 9
                             ! Execute Assignment
next;
FHI1 = hi;
                              ! Phase 1
PHI2 = 10;
                              ! Of Clock Cycle 9
                              ! Execute Pending Assignments
next;
```

```
PHI1 = 10;
                                        ! Phase 2
                                        ! Of Clock Cycle 9
     FHI2 = hi;
     ASN = hi;
                                        ! Deactivate Address Strobe
     LDSN = hi;
                                        ! Neactivate Lower Nata Strobe
     UDSN = hi;
                                        ! Deactivate Upper Data Strobe
     FC = PC + 2;
                                        ! Increment Program Counter
     IR = PFR;
                                        ! Flace Contents Of Prefetch
                                        ! Register Into Instruction
                                        ! Register
     DTACKN = hi;
                                        ! Deactivate Data Transfer
                                        ! Acknowledge(Added)
                                        ! Execute Pending Assignments
     next;
     T = 0
                                        ! JMP (A0)
Jap :=
     ! Phase 1 Of
     PHI1 = hi;
                                        ! Clock Cycle 0
     FHI2 = 10;
                                        ! Data Bus High Impedanced
     DBUS = 0xffff;
                                        ! Memory Read
     RW = hi;
                                        ! Disable Address Bus Buffer
     ADENABLE = lo:
     IIBENABLE = 10;
                                        ! Disable Data Rus Buffer
     IABUS = PC;
                                        ! Place PC On Internal Address
                                        ! Bus
     next;
                                        ! Execute Pending Assignments
     FHI1 = lo;
                                        ! Phase 2 Of
     PHI2 = hi;
                                        ! Clock Cycle 0
                                        ! Enable Address Rus Buffer
     ADENABLE = hi;
     EXARUF = IABUS;
                                        ! Gate Internal Address Bus
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSPACE = 2;
                                        ! Accessing Program
                                        ! Execute Pending Assignments
     ABUS = EXABUF:
                                        ! Address Placed On Rus(Added)
                                        ! Execute Pending Assignments
     next;
     ! Clock Cycle 1
     T = 1;
     next;
                                        ! Execute Assignment
     PHI1 = hi;
                                        ! Phase 1 Of
     PHI2 = 10;
                                        ! Clock Cycle 1
     ASN = lo;
                                        ! Assert Address Strobe
     LIISN = 10;
                                        ! Assert Lower Data Strobe
     UDISN = 10:
                                        ! Assert Upper Data Strobe
```

```
IABUS = A[0];
                              ! Move Jump Address From A[O]
                              ! To Internal Address Buffer
DBENABLE = hi;
                              ! Enable Data Bus
next;
                              ! Execute Pending Assignments
PHI1 = lo;
                              ! Phase 2
PHI2 = hi;
                              ! Of Clock Cycle 1
PC = IABUS;
                              ! Place Jump Address Into Program
                              ! Counter
next;
! Clock Cycle 2
T = 2;
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1
FHI2 = 10;
                              ! Of Clock Cycle 2
                             ! Wait For Memory To Flace
while DTACKN eql hi
                             ! Data On The Bus
                             ! Execute Impending Assignments
    next;
    PHI1 = lo;
                              ! Phase 2
                             ! Of Clock Cycle 2
    PHI2 = hi;
                              ! Execute Assignments
    T = 3;
                              ! Clock Cycle 3
    next;
                              ! Execute Assignment
    PHI1 = hi:
                              ! Phase 1
    PHI2 = 10:
                             ! Of Clock Cycle 3
    [iBUS<15:8> = MEABUS];
                              ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                              ! On Data Bus And
                              ! Asserts DTACKN(Added)
    IITACKN = 10;
                              ! Execute Pending Assignments
     ! Return To Phase 2
    T = 2
                              ! Of Clock Cycle 2
    );
                              ! Execute Impending Assignments
    next;
1 = 3;
                              ! Clock Cycle 3
next;
                              ! Execute Assignment
                              ! Phase 2
FH11 = 10;
                              ! Of Clock Cycle 3
PHI2 = hi:
EXDBUF = DBUS;
                              ! Instruction On Data Bus
                              ! Is Placed In External Data
                              ! Bus Buffer
                              ! Execute Pending Assignments
next;
```

```
T = 4:
                                  ! Clock Cycle 4
next;
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1
PHI2 = 16;
                                  ! Of Clock Cycle 4
next;
PFR = EXDBUF;
                                  ! The Contents Of The External
                                  ! Nata Bus Buffer Are Placed
                                  ! In Prefetch Register
                                  ! Execute Pending Assignments
next;
PHI1 = 10;
                                  ! Phase 2
PHI2 = hi;
                                  ! Of Clock Cycle 4
ASN = hi;
                                  ! Deactivate Address Strobe
LDSN = hi;
                                  ! Deactivate Lower Data Strobe
UDSN = hi;
                                 ! Deactivate Upper Data Strobe
                                 ! Deactivate Data Transfer
DYACKN = hi;
                                  ! Acknowledge(Added)
next;
T = 5;
                                  ! Clock Cycle 5
                                  ! Execute Previous Assignment
next;
PHI1 = hi;
                                  ! Phase 1 Of
                                  ! Clock Cycle 5
PHI2 = 10;
RW = hi;
                                  ! Memory Read
ADENABLE = 10;
                                  ! Disable Address Bus Buffer
                                  ! Disable Data Bus Buffer
DBENABLE = 10:
IABUS = PC;
                                  ! Place PC On Internal Address
                                  ! Bus
next:
                                  ! Execute Fending Assignments
PHI1 = 10;
                                 ! Phase 2 Of
PH12 = hi;
                                  ! Clock Cycle 5
                                  ! Enable Address Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                 ! User Mode
                                 ! Accessing Program
FCSPACE = 2;
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
                                 ! Into External Address Buffer
next;
ABUS = EXABUF;
                                 ! Address Placed On Bus(Added)
next;
                                 ! Execute Pending Assignments
T = 6;
                                  ! Clock Cycle 6
next;
                                  ! Execute Assignment
PHI1 = hi:
                                  ! Phase 1 Of
                                  ! Clock Cycle 6
FHI2 = 10;
ASN = 10;
                                  ! Assert Address Strobe
LISN = lo:
                                  ! Assert Lower Data Strobe
                                  ! Assert Upper Data Strobe
UDSN = 10:
                                  ! Enable Nata Rus
DIBENABLE = h1;
```

```
! Execute Pending Assignments
next;
F'HI1 = 10;
                             ! Phase 2
                             ! Of Clock Cycle 6
PHI2 = hi;
next;
                             ! Execute Pending Assignments
T = 7:
                             ! Clock Cycle 7
next;
                             ! Execute Assignment
PHI1 = hi;
                             ! Phase 1
PHI2 = 10;
                             ! Of Clock Cycle 7
                             ! Wait For Memory To Place
while DTACKN eql hi
    (
                             ! Data On The Bus
                             ! Execute Impending Assignments
    next;
    PHI1 = lo:
                             ! Phase 2
                             ! Of Clock Cycle 7
    FHI2 = hi;
    next:
                             ! Execute Assignments
    \*************************************
                             ! Clock Cycle 8
    T = 8;
                             ! Execute Assignment
    next;
    PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 8
    PHI2 = 10;
    DBUS<15:8> = MEABUS3;
                             ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                             ! On Mata Rus And
                             ! Asserts DTACKN(Added)
    DTACKN = 1g:
                             ! Execute Pending Assignments
    next;
    ! Return To Phase 2
    T = 7
                             ! Of Clock Cycle 7
    );
                         ! Execute Impending Assignments
    next;
T = 8:
                              ! Clock Cycle 8
next;
                              ! Execute Assignment
PHI1 = 10:
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 8
EXDBUF = DBUS;
                             ! Instruction On Data Bus
                             ! Is Placed In External Data
                             ! Bus Ruffer
                              ! Execute Pending Assignments
next:
! Clock Cycle 9
T = 9:
                             ! Execute Assignment
next;
PHI1 = hi;
                             ! Phase 1
```

```
PHI2 = 10;
                                             ! Of Clock Cycle 9
                                             ! The Contents Of The External
     PFR = EXDBUF;
                                             ! Data Bus Buffer Are Placed
                                             ! In Prefetch Register
                                             ! Execute Pending Assignments
     next;
     PHI1 = lo;
                                             ! Phase 2
                                             ! Of Clock Cycle 9
     FHI2 = hi;
     ASN = hi;
                                             ! Deactivate Address Strobe
     LISN = hi;
                                             ! Neactivate Lower Data Strobe
     UDSN = hi;
                                             ! Deactivate Upper Data Strobe
     PC = PC + 2;
                                             ! Increment Program Counter
     IR = PFR;
                                             ! Place Contents Of Prefetch
                                             ! Register Into Instruction
                                             ! Register
                                             ! Deactivate Data Transfer
      DTACKN = hi;
                                             ! Acknowledge(Added)
                                             ! Execute Pending Assignments
      next;
      T = 0
                                             ! Reset Clock Cycle Counter
decode_execute_prefetch :=
                        case IR
                             0155103: add
                                              ! ADD.W D3,D5
                             035001 : moved ! MOVE.W D1,D5
                             032205 : movei ! MOVE.W D5,(A2)
                                              ! JMP (AO) If IR = Octal Value
                             047320 : Jap
                        esac
main :=
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
           decode_execute_prefetch
     )
```

```
/*
                                               */
/*
    MOTOROLA MC68000 MODEL OF THE BED $1000
                                 INSTRUCTION
                                               */
/x
                                               11/
/*
/*
              Structure Declarations
                                               1/
/*
                                               */
state
/*
                                               */
/X
           M68000 Programming Registers
                                               x/
/ X
                                               */
DE0:73<31:0>,
                ! 8 Data Registers
AE0:63<31:0>,
                  ! 7 Address Registers
UA7<31:0>,
                  ! User Stack Pointer
SA7<31:0>,
                  ! System Stack Pointer
PC<31:0>,
                  ! Program Counter -
SR<15:0>,
                  ! .Status Register
/*
                                               */
/*
                                               */
           Temporary Internal Registers
/*
PFR<15:0>,
                  ! Prefetch Register
IR<15:0>,
                  ! Instruction Register
FC<2:0>.
                  ! Function Code Register
EXDBUF<15:0>.
                  ! External Data Bus Buffer Register
EXABUF<23:1>.
                  ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,
                  ! ALU Buffer 1
ALUBUF2<31:0>,
                  ! ALU Buffer 2
DTEMP<15:0>...
                  ! Temporary Data Storage
DISREG<31:0>,
                  ! Temporary Displacement Storage
SKTEMP<15:0>,
                  ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                  ! Temporary Instruction Register Storage
                  ! (Exception Processing)
TEMPADR<31:0>,
                  ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>,
                  ! Temporary Access Type Storage
                  ! (Exception Processing)
                  ! Temporary Vector Address Storage
VECADR<23:0>,
                  ! (Exception Processing)
```

```
HANADR<31:0>,
                          ! Temporary Address Storage For
                           ! Exception Handler Routine
                          ! Clock Cycle Counter
T<7:0>,
RESET,
                        ! Reset Flip-Flop
HALT,
                        ! Halt Flip-Flop
RW.
                       ! Read/Write Flip-Flop
ADENABLE,
                        ! Address Bus Buffer Enable
DBENABLE,
                       ! Data Bus Buffer Enable
ASN,
                        ! Address Strobe Flip-Flop
LUSN,
                        ! Lower Nata Strobe Flip-Flop
UDSN.
                        ! Upper Data Strobe Flip-Flop
DITACKN,
                        ! Nata Transfer Acknowledge Flip-Flop
COUT,
                        ! Carry Flip-Flop
EXCEPT.
                        ! Exception Processing Flip-Flop
READY,
                        ! Ready Flip-Flop
/x
/*
       Model transformation modifications:
                                                                    */
/x
                                                                    */
/*
            1) CDL decoder structure nonexistent in ISP' and un-
                                                                    */
/*
       necessary for model. Eliminated.
                                                                    */
/*
            2) Multi-phase clock structure nonexistent in ISP'.
                                                                    */
/*
       Operations on registers will provide its equivalent.
                                                                    */
/*
            3) Switch structure nonexistent in ISP'. Operation on a
                                                                    */
/x
                                                                    */
       register will provide its equivalent.
/*
            4) The declared bus structures are modeled with registers */
                                                                    */
/x
       without loss of model accurracy. This done to maintain model
                                                                    */
/×
       equivalency and simplicity.
/*
            5) The memory word length was reduced from 16 to 8 bit
                                                                    */
/*
       words to coincide with the ECR's 32-Kbyte memory, to agree with*/
/*
       their PC incrementation, and to enable the use of existing
                                                                    */
/*
       MC68000 assembler and linker/loader models. The memory was
                                                                    */
       also reduced from 8 Mwords to 32 Kbytes.
                                                                    */
14
                                                                    */
/*
       1ABUS<31:0>.
                          ! Internal Address Bus
IDBUS<31:0>,
                          ! Internal Data Bus
SWITCH.
                        ! Power Switch
PHI1.
                        ! Phase 1 Of Two-Phase Clock
PHI2;
                        ! Phase 2 Of Two-Phase Clock
port
/*
                                                                    */
/*
                                                                    */
               External Address and Data Bus
11
                                                                    */
DBUS<15:0>,
                          ! External Data Bus
ABUS<23:1>;
                          ! External Address Bus(changed)
```

```
format
/*
                                                           x/
/*
                 Register Subfields
                                                           */
/x
                                                           */
PCADDR
                       ! Program Counter Address Field
          = PC(23:0),
SKTRACE
                       ! Trace Rit
          = SR<15>,
SAMODE
                       ! Mode Selection Bit
          = SR<13>,
                       ! Curry Bit
SRCARRY
          = SR<0>,
                       ! Overflow bit
SROVER
          = SR<1>,
SKZEKO
          = SR<2>.
                       ! Zero Rit
                       ! Negative Bit
SRNEG
          = SR(3),
SREX
          = SR<4>.
                       ! Extend Bit
SRMASK
          = SR<10:8>,
                       ! Interrupt Mask
FICSPIACE
          = FC<1:0>,
                       ! Memory Access Address Space
          = FC<2>,
FCMODE
                       ! User/Supervisor Mode Bit
FCLOW
          = PC<15:0>,
                       ! PC Low Word
PCHI
          = PC<31:16>,
                       ! PC High Word
DOLWORD
                       ! DEOJ Low Word
          = DE03<15:0>,
                       ! D[1] Low Word
DILWORD
          = DC13<15:0>,
          = BC23<15:0>,
I/2LWORD
                       ! DE23 Low Word
D3LWORD
          = DE33<15:0>,
                       ! D[3] Low Word
II4LWORD
                       ! DE43 Low Word
          = I(4)(15:0),
DSLWORD
                       ! DESI Low Word
          = D(5)(15:0),
                       ! III63 Low Word
DIGL WORD
          = DC63<15:0>,
07LWORD
          = DE71<15:0>,
                       ! DE73 Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW
          = HANADR<15:0>, ! HANADR Low Word
HANAURHI
          = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI
          = TEMPADR<31:16>;! TEMPADR High Word
SICHOTY
/*
                                                           */
/*
                                                           */
                 16K 16-Bit Word Internal Memory
                                                           */
/*
h[0:327673<7:0>;
mac ro
/*
                                                           1/
                                                           */
/*
                Logic Level Macrus
/*
                                                           1/
```

```
= 0 1,
10
    = 1 %,
hi
off
    = 0 1,
on
    = 1 %,
clear = 0 2;
*/
/* Power On and Initialization. This process was not wodeled but is
                                                         */
  added to initialize signals and registers.
                                                         */
/*
                                                         */
power_on_initialize :=
      SWITCH = on:
                                 ! Turn Power On
                                 ! Execute Assignment
      next:
      READY = lo;
                                 ! System Not Ready
      RESET = lo:
                                ! Assert Reset For
                                 ! 100 Miliseconds(Active Low)
      delay(100);
      RESET = hi:
                                ! Deactivate Reset
      next;
                                 ! Execute Pending Assignments
      ASN = hi;
                                ! Initialize Address Strobe
      LDSN = hi;
                                ! Initialize Lower Nata Strobe
      UDSN = hi;
                                ! Initialize Upper Data Strobe
      DITACKN = hi;
                                ! Initialize Data Transfer Acknowledge
      RW = hi;
                                ! Initialize Read/Write(Read On High)
      INUS = Oxffff;
                                ! Place Data Bus In High Impedance State
      MEO \times 100 aJ = O \times ff;
                                ! Place Memory Locations Following The
      M[0x100b] = 0xff;
                                  ! JMP Instruction In A High State
      HALT = hi;
                                ! Initialize Halt Flip-Flop(Active
                                 ! Low)
      T = 0;
                                 ! Initialize Clock Cycle Counter
      READY = hi;
                                 ! System Ready
      /*
                                                         */
      /*
            Routine Initialization Per Hamby and Guillory
                                                         1/
      /*
                                                         */
      ! Set Status Register To User Mode
      SRMUDE = 10;
      D[1] = 0 \times 55555555;
                                ! Place Hex 5555555 Into DC1J
      I(2) = 0.000000000;
                                 ! Place O Into D[2]
      A[0] = 0x1000;
                                 ! Place Hex 1000 Into A[0]
      PC = 0x1000;
                                 ! Place Hex 1000 Into Program Counter
      next
                                 ! Execute Assignments
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                        . */
/* to retrieve modeled instructions for simulation and analysis. It
```

■ 大きからない。 大き回じないというない はいまま でんじん かんじ

(

```
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/# on page VI-15 of their thesis.
fetch_initial_instruction :=
     ! Phase 1 Uf
     PHI1 = hi;
                                      ! Clock Cycle 0
     PHI2 = 10;
     RW = h1;
                                      ! Memory Read
                                      ! Disable Address Bus Buffer
     ADENABLE = lo;
     DRENABLE = 10;
                                      ! Disable Data Bus Buffer
     IABUS = PC;
                                      ! Place PC On Internal Address
                                      ! Rus
     next;
                                      ! Execute Pending Assignments
     PHI1 = lo;
                                      ! Phase 2 Of
                                      ! Clock Cycle 0
     PHI2 = hi:
                                      ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                      ! Gate Internal Address Rus
                                      ! Into External Address Buffer
     FCMODE = SRMODE;
                                      ! User Mode
     FCSPACE = 2;
                                      ! Accessing Program
     next;
                                      ! Execute Impending Assignments
     ABUS = EXABUF;
                                      ! Address Flaced On Bus(Added)
                                      ! Execute Pending Assignments
     next;
     T = 1;
                                      ! Clock Cycle 1
                                      ! Execute Assignment
     next;
                                      ! Phase 1 Of
     PHI1 = hi;
                                      ! Clock Cycle 1
     PHI2 = 10;
                                      ! Assert Address Strobe
     ASN = 10;
                                      ! Assert Lower Data Strobe
     LIISN = lo;
                                      ! Assert Upper Data Strobe
     UIISN = 10;
                                      ! Enable Data Rus
     DBENABLE = hi;
     next;
                                      ! Execute Pending Assignments
     PHI1 = lo;
                                      ! Phase 2
     PHI2 = hi;
                                      ! Of Clock Cycle 1
     next;
                                      ! Execute Pending Assignments
     T = 2;
                                      ! Clock Cycle 2
                                      ! Execute Assignment
     next;
     PHI1 = hi;
                                      ! Phase 1
     PHI2 = 10:
                                      ! Of Clock Cycle 2
```

! Wait For Memory To Place

while DTACKN eql hi

```
! Data On The Bus
     next;
                               ! Execute Impending Assignments
    PHI1 = lo;
                               ! Phase 2
    PHI2 = hi;
                               ! Of Clock Cycle 2
                               ! Execute Assignments
    next;
     ! Clock Cycle 3
                               ! Execute Assignment
     next;
     PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 3
     PHI2 = 10;
     DBUS<15:8> = MCABUSJ;
                               ! Memory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
     DITACKN = 10;
                               ! Asserts DTACKN(Added)
     next;
                               ! Execute Pending Assignments
     ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
     next;
                               ! Execute Impending Assignments
1 = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
FHI1 = 10;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 3
                               ! Instruction On Data Bus
EXDBUF = DBUS;
                               ! Is Placed In External Data
                               ! Bus Buffer
next;
                               ! Execute Pending Assignments
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
FHI2 = 10;
                               ! Of Clock Cycle 4
                               ! The Contents Of The External
PFR = EXDBUF;
                               ! Data Rus Buffer Are Placed
                               ! In Prefetch Register
                               ! Execute Fending Assignments
next;
PHI1 = 10; .
                               ! Phase 2
                               ! Of Clock Cycle 4
PHI2 = hi;
                               ! Deactivate Address Strobe
ASN = hi;
LDSN = hi;
                               ! Deactivate Lower Data Strobe
                               ! Deactivate Upper Data Strobe
UDSN = hi:
IR = PFR;
                               ! Contents Of Prefetch Register
                                ! Are Placed Into Instruction
```

```
! Register
                                    ! Deactivate Nata Transfer(Added)
    DTACKN = hi;
                                    ! Acknowledge
    PC = PC + 2;
                                    ! Increment Program Counter
    next:
                                   ! Execute Pending Assignments
     T = 0
                                    ! Reset Clock Cycle Counter
     )
branch :=
     next;
     T = 9:
                                   ! Clock Cycle 9
     next;
     PHI1 = hi;
     PHI2 = 10;
                                   ! Add Branch Displacement To PC
     PC = PC + IR<7:0>;
     ALIENABLE = 10;
                                   ! Disable Address Bus
     DIBENABLE = 10;
                                   ! Disable Data Rus
     IBUS = Oxffff;
                                   ! Nata Bus High Impedanced
     next;
     PHI1 = lo;
     PHI2 = hi;
     next;
     T = 10;
                                   ! Clock Cycle 10
     next;
     PHI1 = hi:
     PHI2 = 10;
     next;
     PHI1 = lo;
     PHI2 = hi;
     next;
     T = 11;
                                    ! Clock Cycle 11
     next;
                                    ! Phase 1 Of
    PHI1 = hi;
                                    ! Clock Cycle 11
    PHI2 = 10;
    FW = hi;
                                    ! Memory Read
                                    ! Disable Address Bus Buffer
    ADENABLE = 10;
                                    ! Disable Data Bus Buffer
    TIRENABLE = 10;
    DBUS = Oxffff;
                                    ! Data Bus In High Impedance
    1ABUS = PC;
                                    ! Place PC On Internal Address
                                    ! Bus
    next;
                                    ! Execute Pending Assignments
```

```
FHII = 10;
                                ! Phase 2 Of
PHI2 = hi;
                                ! Clock Cycle 11
ADENABLE = hi;
                                ! Enable Address Bus Buffer
EXABUF = IABUS;
                                ! Gate Internal Address Bus
                                ! Into External Address Buffer
FCMODE = SRMODE:
                                ! User Mode
FCSFACE = 2;
                                ! Accessing Program
                                ! Execute Impending Assignments
next;
                                ! Address Placed On Bus(Added)
ARUS = EXABUF:
                                ! Execute Pending Assignments
next;
T = 12:
                                 ! Clock Cycle 12
next;
                                 ! Execute Assignment
                                ! Phase 1 Of
PHI1 = hi:
                                ! Clock Cycle 12
PHI2 = 1o:
                                ! Assert Address Strobe
ASN = 10;
LDSN = lo;
                                ! Assert Lower Data Strobe
UDSN = lo:
                                ! Assert Upper Data Strobe
                                ! Enable Data Bus
DBENABLE = ha;
                                ! Execute Pending Assignments
next:
PHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 12
next:
                                 ! Execute fending Assignments
! Clock Cycle 13
T = 13:
                                 ! Execute Assignment
next;
PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 13
PHI2 = 10;
while RTACKN eql hi
                                ! Wait For Memory To Place .
                                ! Data On The Bus
     (
                                ! Execute Impending Assignments
     next;
     PHI1 = 10;
                               ! Phase 2
                                ! Of Clock Cycle 13
     PHI2 = hi;
     next:
                                ! Execute Assignments
     ! Clock Cycle 14
     T = 14;
                                 ! Execute Assignment
     next;
     PHI1 = hi;
                                 ! Phase 1
                                ! Of Clock Cycle 14
     PHI2 = 10;
     DBUS<15:8> = MCABUSJ;
                                ! Hemory Places Instruction
     DBUS<7:0> = MEABUS + 13;
                                ! On Nata Bus And
                                 ! Asserts DTACKN(Added)
     DTACKN = 10;
                                ! Execute Fending Assignments
     next;
```

```
T = 13
                                 ! Return To Phase 2
                                 ! Of Clock Cycle 13
         );
                                 ! Execute Impending Assignments
         next:
    7 = 14;
                                 ! Clock Cycle 14
                                 ! Execute Assignment
    next;
    PHI1 = lo;
                                 ! Phase 2
    PHI2 = hi;
                                 ! Of Clock Cycle 14
    EXDRUF = DRUS;
                                 ! Instruction On Data Rus
                                 ! Is Flaced In External Data
                                 ! Bus Buffer
    next;
                                 ! Execute Pending Assignments
    T = 15;
                                 ! Clock Cycle 15
    next;
                                 ! Execute Assignment
                                 ! Phase 1
    PHI1 = hi;
                                 ! Of Clock Cycle 15
    PHI2 = 10;
    PFR = EXDBUF;
                                 ! The Contents Of The External
                                 ! Data Bus Buffer Are Placed
                                 ! In Prefetch Register
                                 ! Execute Pending Assignments
    next;
    PHI1 = lo;
                                 ! Phase 2
                                 ! Of Clock Cycle 15
    PHI2 = hi;
    ASN = hi;
                                 ! Deactivate Address Strobe
    IR = PFR;
                                 ! Place Frefetch Register Into IR
                                ! Activate Data Transfer Acknowledge
    DTACKN = hi;
    LIISN = hi:
                                 ! Deactivate Lower Data Strobe
    UDSN = hi;
                                 ! Deactivate Upper Data Strobe
    PC = PC + 2;
                                ! Increment Program Counter
    next;
                                 ! Execute Pending Assignments
    T = 0
                                 ! Reset Clock Cycle Counter
nobranch :=
      PC = PC + 2;
                               ! Incrment PC
      next;
    T = 5;
                             ! Clock Cycle 5
      next;
      PHI1 = hi;
      PHI2 = lo:
```

```
ADENABLE = 10;
                        ! Disable Address Bus
 DBENABLE = lo;
                        ! Disable Data bus
 DBUS = 0xffff;
                        ! Data Bus High Impedanced
 next;
 PHI1 = lo;
 PHI2 = hi;
 next;
! Clock Cycle 6
 next;
 PHI1 = hi;
 PHI2 = 10;
 next;
 PHI1 = lo:
 PHI2 = hi;
 next;
T = 7;
                      ! Clock Cycle 7
 next;
 PHI1 = hi;
 PHI2 = 10;
 next;
 PHI1 = 10;
 PH12 = h1;
 next;
! Clock Cycle 8
 T = 8;
 next;
 PHI1 = hi;
 PHI2 = 10;
 next:
 PHI1 = lo;
 PHI2 = hi;
 IR = PFR;
                      ! Place Prefetch Register Into IR
 next;
T = 0;
                      ! Reset Clock Cycle Counter
                         ! REQ $1000
```

beg

```
FHI1 = hi;
                                 ! Phase 1 Of
                                 ! Clock Cycle O
PRI2 = 10;
RW = hi;
                                 ! Memory Read
ADENABLE = 10;
                                 ! Disable Address Bus
DRENABLE = 10;
                                 ! Disable Data Bus
                                   ! Nata Bus In High Impedance State
DBUS = 0xffff;
IABUS = PC;
                                 ! Place PC On Internal Address
                                 ! Execute Pending Assignments
next;
                                 ! Phase 2 Of
FHI1 = lo;
                                 ! Clock Cycle 0
PHI2 = hi:
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
                                 ! Into External Address Buffer
FCMODE = SRMODE;
                                 ! User Mode
FCSPACE = 2;
                                 ! Accessing Program
                                 ! Execute Impending Assignments
next:
ABUS = EXABUF;
                                 ! Address Placed On Bus(Added)
                                 ! Execute Pending Assignments
next:
T = 1;
                                 ! Clock Cycle 1
next;
                                 ! Execute Assignment
                                 ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                                 ! Clock Cycle 1
                                 ! Assert Address Strobe
ASN = lo;
                                 ! Assert Lower Data Strobe
LDSN = lo;
UIISN = lo:
                                 ! Assert Upper Data Strobe
DBENABLE = hi:
                                 ! Enable Data Bus
                                 ! Execute Pending Assignments
next;
PHI1 = lo:
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 1
next;
                                 ! Execute Pending Assignments
7 = 2;
                                 ! Clock Cycle 2
                                 ! Execute Assignment
next;
                                 ! Phase 1
PHI1 = hi;
PHI2 = 10;
                                 ! Of Clock Cycle 2
while DTACKN eql hi
                                 ! Wait For Memory To Place
                                 ! Data On The Bus
                                 ! Execute Impending Assignments
     next;
                                 ! Phase 2
     PHI1 = lo;
     PHI2 = hi:
                                 ! Of Clock Cycle 2
                                 ! Execute Assignments
     nexti
```

```
T = 3;
                               ! Clock Cycle 3
    next;
                               ! Execute Assignment
     PHI1 = hi;
                               ! Phase 1
    PHI2 = 10;
                               ! Of Clock Cycle 3
     DBUS<15:8> = MCABUS];
                               ! Memory Flaces Instruction
    DBUS<7:0> = MEABUS + 13;
                               ! On Jiata Bus And
                               ! Asserts DTACKN(Added)
     DTACKN = 10;
    next:
                               ! Execute Pending Assignments
     ! Return To Phase 2
                               ! Of Clock Cycle 2
    );
                              ! Execute Impending Assignments
    next;
T = 3:
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
FHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
                               ! Instruction On Data Rus
EXDBUF = DRUS;
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Fending Assignments
next;
T = 4;
                               ! Clock Cycle 4
                               ! Execute Assignment
next;
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 4
PHI2 = 10:
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
PHII = lo:
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 4
DTACKN = hi;
                               ! Deactivate Data Transfer(Added)
ASN = hi;
                             ! Deactivate Address Strobe
                             ! Deactivate Lower Data Strobe
LDSN = hi;
UNSN = hi;
                             ! Deactivate Upper Data Strobe
                               ! Acknowledge
next;
                               ! If Zero Rit 1s Set Then
if SRZERO
                               ! Execute Branch Cycle
  branch
                               ! Else Execute Nonbranch Cycle
else nobranch
```

! MOVE.W D1,D3 and MOVE.W D2,D3

move :=

```
! Phase 1 Of
PHI1 = hi;
                                   ! Clock Cycle 0
PHI2 = 10:
                                   ! Place Data Bus In High Impedance
DBUS = 0xffff;
RW = hi;
                                  ! Memory Read
ADENABLE = 10:
                                  ! Disable Address Bus Buffer
DIBENABLE = 10;
                                  ! Disable Data Bus Buffer
IABUS = PC;
                                  ! Place PC On Internal Address
                                  ! Rus
case Iƙ
                                   ! Place Low Word From DE13/BE23 Onto
    0x3601: IDBUS = D1LWORD
                                   ! Internal Data Bus
    0x3602: INBUS = D2LWORD
esac;
                                   ! Execute Pending Assignments
next;
                                   ! Phase 2 Of
PHI1 = lo;
                                   ! Clock Cycle 0
PHI2 = hi;
                                   ! Enable Address Bus Buffer
ADENABLE = hi;
                                  ! Gate Internal Address Bus
EXABUF = IABUS;
                                  ! Into External Address Buffer -
FCMODE = SRMODE;
                                  ! User Mode
FCSPACE = 2;
                                  ! Accessing Program
                                  ! Clear Status Register Carry Bit
SRCARRY = lo;
                                  ! Clear Status Register Overflow Bit
SROVER = 10;
                                  ! Clear Status Register Zero Bit
SRZERO = lo:
SRNEG = 10;
                                  ! Clear Status Register Negative Bit
                                  ! Place Data From Internal Data Bus
D3LWORD = IDBUS;
                                  ! Into Low Word Of D[3]
                                  ! Execute Impending Assignments
next;
ABUS = EXABUF;
                                  ! Address Placed On Bus(Added)
next:
                                   ! Execute Pending Assignments
T = 1:
                                   ! Clock Cycle 1
                                   ! Execute Assignment
next;
                                   ! Phase 1 Of
PHI1 = hi;
                                   ! Clock Cycle 1
PH12 = 10;
ASN = lo;
                                   ! Assert Address Strobe
LIISN = lo;
                                  ! Assert Lower Data Strobe
                                  ! Assert Upper Data Strobe
UDSN = 10;
                                  ! Enable Data Rus
DBENABLE = hi;
                                  ! Set Status Register Zero Bit
if D3LWORD eql 0
                                  ! If Moved Data Is Zero
  SRZERO = hi:
                                  ! Execute Pending Assignments
next;
PH11 = 1c;
                                   ! Phase 2
PHI2 = hi;
                                   ! Of Clock Cycle 1
if DC23<15>
                                   ! Set Status Register Negative
```

! Bit If Moved Data Is Negative

SRNEG = hi:

```
! Execute Pending Assignments
next;
! Clock Cycle 2
next;
                              ! Execute Assignment
PHI1 = hi;
                             ! Phase 1
PHI2 = 10;
                             ! Of Clock Cycle 2
                             ! Wait For Memory To Place
while DTACKN eql hi
                             ! Data On The Bus
                             ! Execute Impending Assignments
    next;
    PHI1 = lo:
                             ! Phase 2
                             ! Of Clock Cycle 2
    PHI2 = hi;
    next;
                              ! Execute Assignments
     T = 3;
                             ! Clock Cycle 3
    next;
                             ! Execute Assignment
    PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 3
    PHI2 = lo;
    DBUS<15:8> = MCABUSD;
                             ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                            ! On Data Bus And
                             ! Asserts DTACKN(Added)
    DTACKN = lo;
                             ! Execute Pending Assignments
    next;
     /*****************************
                              ! Return To Phase 2
    T = 2
                             ! Of Clock Cycle 2
     );
                             ! Execute Impending Assignments
    next;
T = 3:
                             ! Clock Cycle 3
next;
                             ! Execute Assignment
                             ! Phase 2
PHI1 = lo;
                              ! Of Clock Cycle 3
PHI2 = hi;
                             ! Instruction On Data Rus
EXDRUF = DRUS;
                              ! Is Placed In External Data
                             ! Bus Buffer
                             ! Execute Pending Assignments
next;
T = 4;
                             ! Clock Cycle 4
                             ! Execute Assignment
next;
                              ! Phase 1
PHI1 = hi;
PHI2 = 10;
                              ! Of Clock Cycle 4
                             ! The Contents Of The External
PFR = EXDBUF;
                             ! Data Bus Buffer Are Placed
                             ! In Prefetch Register
```

```
! Execute Pending Assignments
     next;
     PHI1 = 10:
                                         ! Phase 2
     PHI2 = hi;
                                         ! Of Clock Cycle 4
                                         ! Deactivate Address Strobe
     ASN = hi;
                                         ! Neactivate Lower Nata Strobe
     LUSN = hi:
                                         ! Deactivate Upper Data Strobe
     UDSN = hi;
                                         ! Contents Of Prefetch Register
     IR = PFR;
                                         ! Are Flaced Into Instruction
                                         ! Register
     DTACKN = hi;
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
     PC = PC + 2;
                                        ! Increment Program Counter
                                        ! Execute Impending Assignments
     next:
     T = 0
                                         ! Reset Clock Cycle Counter
                                         ! JMP (A0)
Jmp :=
     PHI1 = hi;
                                         ! Phase 1 Of
     FHI2 = 10;
                                         ! Clock Cycle 0
     DBUS = Oxffff;
                                         ! Place Data Bus In A High Impedance
     KW = hi;
                                         ! Memory Read
     ADENABLE = 10;
                                         ! Disable Address Bus Buffer
                                         ! Disable Data Bus Ruffer
     IMENABLE = 10;
                                         ! Place PC On Internal Address
     IARUS = PC;
     next;
                                         ! Execute Pending Assignments
                                         ! Phase 2 Of
     PHI1 = 10:
     PHI2 = hi;
                                         ! Clock Cycle O
                                         ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
     FCMODE = SRMODE:
                                         ! User Mode
     FCSPACE = 2;
                                         ! Accessing Program
                                         ! Execute Pending Assignments
     next:
     ABUS = EXABUF;
                                         ! Address Placed On Bus(Added)
     next;
                                         ! Execute Pending Assignments
     T = 1;
                                         ! Clock Cycle 1
     next;
                                         ! Execute Assignment
     PHI1 = hi;
                                         ! Phase 1 Of
     PHI2 = 10:
                                         ! Clock Cycle 1
                                         ! Assert Address Strobe
     ASN = 10;
                                         ! Assert Lower Data Strobe
     LDSN = lo;
```

```
UDSN = 10:
                                  ! Assert Upper Data Strobe
 IABUS = ACOD:
                                  ! Move Jump Address From ALO]
                                 ! To Internal Address Buffer
 DBENABLE = hi:
                                 ! Enable Nata Rus
next;
                                  ! Execute Pending Assignments
PHI1 = 10:
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 1
PC = IABUS:
                                 ! Place Jump Address Into Program
                                 ! Counter
next:
T = 2;
                                 ! Clock Cycle 2
next:
                                 ! Execute Assignment
PHI1 = hi;
                                ! Phase 1
PHI2 = 10:
                                ! Of Clock Cyrle 2
while DTACKN eql hi
                                ! Wait For Memory To Place
                                ! liata On The Rus
                                ! Execute Impending Assignments
     next;
     PHI1 = 10;
                                ! Phase 2
     PHI2 = hi;
                                ! Of Clock Cycle 2
     next;
                                 ! Execute Assignments
     T = 3;
                                ! Clock Cycle 3
     next;
                                ! Execute Assignment
                                ! Phase 1
     PHI1 = hi;
     PHI2 = 10
                                 ! Of Clock Cycle 3
     DBUS<15:8> = MEARUS3;
                                ! Memory Places Instruction
     DRUS<7:0> = MEABUS + 13;
                                ! On Data Bus And
                                ! Asserts DTACKN(Added)
     DTACKN = lo;
     next;
                                 ! Execute Pending Assignments
     T = 2
                                 ! Return To Phase 2
                                 ! Of Clock Cycle 2
     );
                                 ! Execute Impending Assignments
     next;
T = 3:
                                 ! Clock Cycle 3
next:
                                 ! Execute Assignment
                                 ! Phase 2
PHI1 = lo;
PHI2 = hi;
                                 ! Of Clock Cycle 3
EXBBUF = DBUS;
                                ! Instruction On Data Bus
                                 ! Is Placed In External Data
                                 ! Rus Buffer
next;
                                 ! Execute Pending Assignments
```

```
T = 4;
                                  ! Clock Cycle 4
next;
                                  ! Execute Assignment
PHI1 = hi:
                                  ! Phase 1
PHI2 = 10:
                                  ! Of Clock Cycle 4
next;
FFR = EXDRUF:
                                  ! The Contents Of The External
                                 ! Data Bus Buffer Are Placed
                                  ! In Prefetch Register
                                  ! Execute fending Assignments
next;
PHI1 = lo;
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 4
ASN = hi;
                                 ! Neactivate Address Strobe
LDSN = hi;
                                 ! Deactivate Lower Data Strobe
                                 ! Deactivate Upper Data Strobe
UDSN = hi;
                                 ! Deactivate Data Transfer
DITACKN = hi;
                                 ! Acknowledge(Added)
! Clock Cycle 5
T = 5;
                                 ! Execute Previous Assignment
next;
                                 ! Phase 1 Of
PHI1 = hi;
                                 ! Clock Cycle 5
fHI2 = 10;
                                 ! Memory Read
RW = hi;
                                 ! Disable Address Bus Buffer
ADENABLE = 10;
                                 ! Disable Data Rus Buffer
DEENABLE = 10;
                                 ! Place PC On Internal Address
IABUS = PC;
                                  ! Bus
                                 ! Execute Pending Assignments
next;
                                 ! Phase 2 Of
FKII = 10;
                                 ! Clock Cycle 5
PHI2 ≈ hi;
                                 ! Enable Address Bus Buffer
ADENABLE = hi;
                                 '! User Mode
FCMODE = SRMODE;
                                 ! Accessing Program
FCSPACE = 2;
                                 ! Gate Internal Address Rus
EXABUF = IABUS;
                                 ! Into External Address Buffer
next;
                                 ! Address Placed On Bus(Added)
ABUS = EXABUF;
                                  ! Execute Pending Assignments
next;
! Clock Cycle 6
T = 6:
                                  ! Execute Assignment
next;
                                  ! Phase 1 Of
PHI1 = hi:
                                  ! Clock Cycle 6
PHI2 = 10;
                                  ! Assert Address Strobe
ASN = 10;
                                  ! Assert Lower Data Strobe
LIISH = 10;
                                  ! Assert Upper Data Strobe
UDSN = lo;
```

```
DBENABLE = hi:
                              ! Enable Data Bus
next:
                              ! Execute Pending Assignments
PHI1 = 10:
                              ! Phase 2
PHI2 = hi;
                              ! Of Clock Cycle 6
next;
                              ! Execute Pending Assignments
next;
                             ! Execute Assignment
PHI1 = hi;
                             ! Phase 1
PHI2 = 10;
                             ! Of Clock Cycle 7
while DTACKN eql hi
                             ! Wait For Memory To Place
     (
                             ! Note On The Kus
    next;
                             ! Execute Impending Assignments
    FHI1 = lo;
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Eycle 7
    next;
                             ! Execute Assignments
    T = 81
                             ! Clock Cycle 8
    next;
                             ! Execute Assignment
    PHI1 = hi;
                            ! Phase 1
    PHI2 = 16:
                             ! Of Clock Cycle 8
    DBUS<15:8> = MEABUS];
                             ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 1];
                             ! On Data Bus And
    INTACKN = 16:
                             ! Asserts DTACKN(Added)
    next:
                             ! Execute Pending Assignments
    T = 7
                            ! Return To Phase 2
                            ! Of Clock Cycle 7
    );
                            ! Execute Impending Assignments
    next;
T = 8;
                             ! Clock Cycle B
next:
                             ! Execute Assignment
PHI1 = 10:
                             ! Phase 2
PHI2 = hi;
                             ! Of Clock Cycle 8
EXDBUF = DBUS:
                             ! Instruction On Data Bus
                             ! Is Placed In External Data
                             ! Bus Buffer
                             ! Execute Pending Assignments
next:
T = 9;
                             ! Clock Cycle 9
next;
                             ! Execute Assignment
```

```
PHI1 = hi;
                                             ! Phase 1
                                             ! Of Clock Cycle 9
     PHI2 = 10;
                                             ! The Contents Of The External
     PFR = EXDBUF;
                                             ! Data Bus Buffer Are Placed
                                             ! In Prefetch Register
                                             ! Execute Pending Assignments
     next;
                                             ! Phase 2
      PHI1 = 10;
     PHI2 = hi;
                                             ! Of Clock Cycle 9
                                             ! Deactivate Address Strobe
      ASN = hi;
     LISN = hi;
                                             ! Neactivate Lower Data Strobe
                                             ! Deactivate Upper Data Strobe
     UDSN = hi;
                                             ! Increment Program Counter
     FC = FC + 2;
      IR = PFR;
                                             ! Place Contents Of Prefetch
                                             ! Register Into Instruction
                                             ! Register
                                             ! Deactivate Data Transfer
      DTACKN = hi;
                                             ! Acknowledge(Added)
                                             ! Execute Pending Assignments
     next:
                                             ! Reset Clock Cycle Counter
      T = 0
decode_execute_prefetch :=
                        case IR
                                                   ! BEQ $1000
                             0x67fc,0x67f8: beq
                             0x3601,0x3602; move ! MOVE.W D1,D3 and D2,D3
                                                   ! JMP (A0) If IR = Octal Value
                                     Ox4edO: jmp
                        esuc
main :=
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
           decode_execute_prefetch
```

)

```
/x
                                                 */
/#
    MOTOROLA MC68000 MODEL OF THE BTST III, (A1) INSTRUCTION
                                                 ¥/
/*
                                                 1/
/*
                                                 */
/*
              Structure Declarations
                                                 */
/*
*/
                                                 */
/*
           M68000 Programming Registers
                                                 1/
DE0:73<31:0>,
                   ! 8 Data Registers
AE0:63<31:0>,
                   ! 7 Address Registers
UA7<31:0>,
                   ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Fointer
PC<31:0>,
                   ! Program Counter
SR<15:0>,
                   ! Status Register
/*
                                                 */
/*
           Temporary Internal Registers
                                                 */
/*
! Prefetch Register
PFR<15:0>,
IR<15:0>,
                   ! Instruction Register
                   ! Function Code Register
FC<2:0>,
                   ! External Data Bus Buffer Register
EXDBUF<15:0>,
                   ! External Address Bus Buffer Register(changed)
EXABUF<23:1>,
                   ! ALU Buffer 1
ALUBUF1<31:0>,
ALUBUF2<31:0>,
                   ! ALU Buffer 2
DTEMP<15:0>,
                   ! Temporary Data Storage
                   ! Temporary Displacement Storage
DISREG<31:0>,
                   ! Temporary Status Register Storage
SRTEMP<15:0>,
                   ! (Exception Processing)
                   ! Temporary Instruction Register Storage
IRTEMP<15:0>,
                   ! (Exception Processing)
                   ! Temporary Cycle Address Storage
TEMPADR<31:0>,
                   ! (Exception Processing)
ACTYPE<15:0>,
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
                   ! Temporary Vector Address Storage
VECADR<2310>,
                   ! (Exception Processing)
```

```
HANADR<31:0>.
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
T<7:0>,
                        ! Clock Cycle Counter
HOLD<3:0>.
                        ! Temorary Holding Register
RESET,
                     ! Reset Flip-Flop
HALT.
                     ! Halt Flip~Flop
RW.
                     ! Read/Write Flip-Flop
ABENABLE.
                      ! Address Bus Buffer Enable
DBENABLE,
                      ! Data Bus Buffer Enable
ASN,
                      ! Address Strobe Flip-Flop
LUSN,
                     ! Lower Data Strobe Flip-Flop
UDSN.
                     ! Upper Data Strobe Flip-Flop
LITACKN.
                     ! Date Transfer Acknowledge Flip-Flop
COUT,
                     ! Carry Flip-Flop
EXCEPT,
                     ! Exception Processing Flip-Flop
READY,
                     ! Ready Flip-Flop
/ ¥
                                                              x/
                                                              */
/*
       Model transformation modifications:
/*
                                                              */
/x
           1) CDL decoder structure nonexistent in ISP' and un-
                                                              */
       necessary for model. Eliminated.
                                                              */
/*
/ x
           2) Multi-phase clock structure nonexistent in ISP'.
                                                              */
       Operations on registers will provide its equivalent.
                                                              */
/×
           3) Switch structure nonexistent in ISP'. Operation on a
                                                              1./
/*
/x
       register will provide its equivalent.
                                                              */
           4) The declared bus structures are modeled with registers */
/*
       without loss of model accurracy. This done to maintain model
                                                              */
/*
                                                              1/
       equivalency and simplicity.
/*
                                                              */
           5) The memory word length was reduced from 16 to 8 bit
/*
       words to coincide with the ECR's 32-Kbyte memory, to agree with#/
/*
                                                              */
/*
       their PC incrementation, and to enable the use of existing
/ X
       MC68000 assembler and linker/loader models. The memory was
                                                              1/
                                                              */
/×
       also reduced from 8 Mwords to 32 Kbytes.
/*
! Internal Address Bus
IABUS<31:0>,
                        ! Internal Data Bus
1DBUS<31:0>.
                     ! Power Switch
SWITCH,
                     ! Phase 1 Of Two-Phase Clock
FHI1,
                     ! Phase 2 Of Two-Phase Clock
PHI2;
port
/*
                                                              x/
                                                              */
/*
              External Address and Data Bus
/*
                                                              1/
! External Data Bus
DBUS<15:0/,
```

```
ABUS<23:1>;
                     ! External Address Bus(changed)
format
/x
                                                        x/
/*
                                                        */
                Register Subfields
/x
                                                        */
PCADDR
                      ! Program Counter Address Field
         = PC<23:0>;
SRTRACE
                      ! Trace Rit
         = SR<15>,
                      ! Mode Selection Bit
SRMODE
         = S8<13>.
SRCARRY
         = SR<0>.
                      ! Carry Bit
SROVER
         = SR<1>,
                      ! Overflow Bit
SKZERO
         = SR<2>
                      ! Zero kit
SKNEG
         = SR<3>,
                      ! Negative Bit
SREX
         = SR<4>,
                      ! Extend Bit
SRMASK
         = SR<10:8>
                      ! Interrupt Mask
FCSPACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCMODE
         = FC<2>,
                      ! User/Supervisor Mode Bit
FCLOW
                      ! PC Low Word
         = PC(15:0),
PCHI
         = PC<31:16>,
                      ! PC High Word
         = DE03<15:0>,
DOLWORD
                      ! II[0] Low Word
                      ! D[1] Low Word
DILWORD
         = DC13<15:0>,
D2LWORD
                      ! DE23 Low Word
         = D[2]<15:0>,
D3LWORD
         = D(3)(15:0),
                      ! D[3] Low Word
II4LWORD
         = D[4](15:0),
                      ! II[4] Low Word
                      ! D[5] Low Word
DSLWORD
         = D(5)(15:0),
                      ! IIE63 Low Word
DI6LWORD
         = B[6]<15:0>,
         = DE73<15:0>,
                      ! D[7] Low Word
D7LWORD
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW
        = HANAUR<15:0>, ! HANAUR Low Word
         = HANAUR<31:16>,! HANAUR High Word
HANADRHI
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
         = TEMPADR<31:16>;! TEMPADR High Word
TEMPAURHI
memory
/*
                                                        x/
                16K 16-Bit Word Internal Memory
                                                        */
/*
                                                        */
/*
MEG:327673<7:0>:
macro
/*
                                                        */
/1
               Logic Level Macros
```

```
10
     = 0 1,
hi
     = 1 1,
off
     = 0 %,
     = 1 %,
CID
clear = 0 %;
/*
                                                            */
  Power On and Initialization. This process was not modeled but is
/*
   added to initialize signals and registers.
                                                            1/
                                                            */
/*
power_on_initialize :=
                                   ! Turn Power On
      SWITCH = on;
      next;
                                   ! Execute Assianment
      READY = lo;
                                   ! System Not Ready
                                   ! Assert Reset For
      RESET = 10;
                                   ! 100 Miliseconds(Active Low)
      delay(100);
                                   ! Deactivate Reset
      RESET = hi;
      next;
                                   ! Execute Pending Assignments
      ASN = hi;
                                   ! Initialize Address Strobe
      LDSN = hi;
                                   ! Initialize Lower Nata Strobe
                                  ! Initialize Upper Nata Strobe
      UDSN = hi;
      DTACKN = hi;
                                  ! Initialize Data Transfer Acknowledge
      RW = hi;
                                  ! Initialize Read/Write(Read On High)
      IIBUS = 0xffff;
                                  ! Place Data Rus In High Impedance State
      ME0 \times 1008] = 0 \times ff;
                                  ! Place Memory Locations Following The
      ME0\times1009] = 0\times ff;
                                   ! JMP Instruction In A High State
      HALT = hi;
                                   ! Initialize Halt Flip-Flop(Active
                                   ! Low)
      T = 0;
                                   ! Initialize Clock Cycle Counter
      READY = hi;
                                   ! System Ready
       /*
                                                            */
      /*
            Routine Initialization Per Hamby and Guillary
                                                            1/
                                                            */
      /*
       SEMODE = lo;
                                   ! Set Status Register To User Mode
                                   ! Place Hex 3 Into D[1]
      I([1] = 0x3;
      DC23 = 0x555555555;
                                   ! Place Hex 5555555 Into D[2]
      A[0] = 0x1000;
                                   ! Place Hex 1000 Into ACO]
      A[1] = 0 \times 2001;
                                   ! Place Hex 2001 Into A[1]
                                   ! Initialize Location 2000 To Zero
      M[0\times2000] = 0\times0;
                                  ! Initialize Location 2001 To Hex 55
      M[0x2001] = 0x55;
      PC = 0x1000;
                                  ! Place Hex 1000 Into Program Counter
                                   ! Execute Assignments
      next
```

)

```
/*
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                         */
/* to retrieve modeled instructions for simulation and analysis. It
                                                        */
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
                                                        x/
/* on page VI-15 of their thesis.
/*
                                                         */
fetch_initial_instruction :=
    PHI1 = hi;
                                   ! Phase 1 Of
    PHI2 = 10;
                                   ! Clock Cycle 0
    RW = hi;
                                   ! Memory Read
    ADENABLE = 10:
                                   ! Disable Address Bus Buffer
    DRENABLE = 10;
                                   ! Disable Data Bus Buffer
    IABUS = PC;
                                   ! Place PC On Internal Address
                                   ! Execute Pending Assignments
    next;
                                   ! Phase 2 Of
    PHI1 = 10;
    PHI2 = hi;
                                   ! Clock Cycle 0
                                   ! Enable Address Bus Buffer
    ANENABLE = hi;
    EXABUF = IABUS;
                                   ! Gate Internal Address Bus
                                   ! 'Into External Address Buffer
    FCMODE = SRMODE;
                                   ! User Mode
    FCSFACE = 2;
                                   ! Accessing Program
                                   ! Execute Impending Assignments
    next;
    ABUS = EXABUF;
                                   ! Address Flaced On Bus(Added)
    next:
                                   ! Execute Pending Assignments
    T = 1;
                                   ! Clock Cycle 1
    next;
                                   ! Execute Assignment
    PHI1 = hi;
                                   ! Phase 1 Of
    PH12 = 10:
                                   ! Clock Cycle 1
    ASN = lo:
                                   ! Assert Address Strobe
    LUSN = 10;
                                   ! Assert Lower Data Strobe
                                   ! Assert Upper Data Strobe
    UDSN = lo:
    DBENABLE = hi:
                                   ! Enable Data Rus
    next;
                                   ! Execute Pending Assignments
                                   ! Phase 2
    PHI1 = 10;
    PHI2 = hi;
                                   ! Of Clock Cycle 1
                                   ! Execute Fending Assignments
    next;
    T = 2;
                                   ! Clock Cycle 2
```

! Execute Assignment

next;

```
PHI1 = hi:
                              ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 2
while DTACKN eql hi
                               ! Wait For Memory To Place
                               ! Data On The Bus
                               ! Execute Impending Assignments
     next;
     PHI1 = lo;
                              ! Phase 2
     PH12 = hi;
                              ! Of Clock Cycle 2
                              ! Execute Assignments
     next;
     ! Clock Cycle 3
     next;
                               ! Execute Assignment
                               ! Phase 1
     PHI1 = hi;
     PHI2 = 10;
                               ! Of Clock Cycle 3
     DBUS<15:8> = MEABUS3;
                              ! Memory Places Instruction
     DBUS<7:0> = MEARUS + 13;
                              ! On Data Bus And
     DTACKN = lo;
                              ! Asserts DTACKN(Added)
                               ! Execute Fending Assignments
     next:
     ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
     next;
                           ! Execute Impending Assignments
/**************************
T = 3:
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
                               ! Phase 2
PHI1 = lo;
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
next;
                               ! Execute Pending Assignments
T = 4:
                               ! Clock Cycle 4
                               ! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi;
                               ! Of Clock Cycle 4
PHI2 = 10;
                               ! The Contents Of The External
PFR = EXDBUF;
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
                               ! Execute Fending Assignments
next;
                               ! Phase 2
PHII = lo;
                               ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                               ! Deactivate Address Strobe
```

```
LDSN = hi:
                                         ! Neactivate Lower Data Strobe
     UDSN = hi:
                                         ! Deactivate Upper Data Strobe
     IR = PFR;
                                         ! Contents Of Prefetch Register
                                         ! Are Placed Into Instruction
                                         ! Register
     DTACKN = hi;
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
     PC = PC + 2;
                                         ! Increment Program Counter
     next;
                                         ! Execute Pending Assignments
     T = 0
                                         ! Reset Clock Cycle Counter
     )
                                         ! BTST [01,(A1)
btst :=
     PHI1 = hi;
                                         ! Phase 1 Of
     PHI2 = 10;
                                         ! Clock Cycle 0
     RW = hi;
                                         ! Memory Read
     ADENABLE = 10;
                                         ! Disable Address Bus Buffer
     DBUS = 0xffff;
                                         ! Data Bus High Impedanced
     DRENABLE = lo;
                                         ! Disable Data Bus Buffer
                                         ! Place PC On Internal Address
     IABUS = PC;
                                         ! Rus
                                         ! Execute Pending Assignments
     next;
     FHI1 = lo;
                                         ! Phase 2 Of
     PHI2 = hi:
                                         ! Clock Cycle O
                                        ! Enable Address Bus Buffer
     ABENABLE = hi;
     EXABUF = IABUS;
                                        ! Gate Internal Address Bus
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                         ! User Mode
     FCSPACE = 2;
                                         ! Accessing Program
     next;
                                        ! Execute Impending Assignments
     ARUS = EXABUF;
                                         ! Address Placed On Bus(Added)
     next;
                                         ! Execute Pending Assignments
     T = 1;
                                         ! Clock Cycle 1
     next;
                                         ! Execute Assignment
     PHI1 = hi;
                                         ! Phase 1 Of
     PH12 = 10;
                                         ! Clock Cycle 1
     ASN = 10;
                                         ! Assert Address Strobe
     LISN = 10;
                                         ! Assert Lower Data Strobe
                                         ! Assert Upper Data Strobe
     UDSN = 10;
     IBENABLE = hi;
                                         ! Enable Data Bus
     next;
                                         ! Execute Pending Assignments
                                         ! Phase 2
     PHI1 = lo;
     PHI2 = hi;
                                         ! Of Clock Cycle 1
                                         ! Execute Fending Assignments
     next;
```

```
T = 2;
                             ! Clock Cycle 2
next;
                             ! Execute Assignment
PHI1 = hi;
                             ! Phase 1
PHI2 = 10;
                             ! Of Clock Cycle 2
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Data On The Bus
    next;
                             ! Execute Impending Assignments
    PHI1 = lo;
                             ! Phase 2
                             ! Of Clock Cycle 2
    PHI2 = hi;
                             ! Execute Assignments
    nexti
    T = 3;
                             ! Clock Cycle 3
    next;
                             ! Execute Assignment
    PHI1 = hi:
                             ! Phase 1
                             ! Of Clock Cycle 3
    PHI2 = 10;
    DBUS<15:8> = MEABUS];
                             ! Memory Places Instruction
                             ! On Data Bus And
    DBUS<7:0> = MCABUS + 1];
                             ! Asserts DTACKN(Added)
    DTACKN = 1o:
    next;
                             ! Execute Pending Assignments
    ! Return To Phase 2
    T = 2
                             ! Of Clock Cycle 2
    );
    next:
                             ! Execute Impending Assignments
T = 3;
                             ! Clock Cycle 3
                             ! Execute Assignment
next;
                             ! Phase 2
PHI1 = lo;
                             ! Of Clock Cycle 3
PHI2 = hi;
                             ! Instruction On Data Bus
EXDRUF = DBUS;
                             ! Is Placed In External Data
                             ! Bus Buffer
                             ! Execute Fending Assignments
next;
T = 4;
                             ! Clock Cycle 4
next;
                             ! Execute Assignment
                             ! Phase 1
PHI1 = hi;
PHI2 = 10;
                             ! Of Clock Cycle 4
PFR = EXDBUF;
                             ! The Contents Of The External
                             ! Data Rus Buffer Are Placed
                             ! In Prefetch Register
                             ! Execute Pending Assignments
```

next;

```
! Phase 2
PHI1 = lo;
                                   ! Of Clock Cycle 4
PHI2 = hi;
                                   ! Deactivate Address Strobe
ASN = hi;
                                   ! Deactivate Lower Data Strobe
LDSN = hi:
                                   ! Deactivate Upper Data Strobe
UDSN = hi;
                                   ! Register
DTACKN = hi:
                                   ! Deactivate Data Transfer(Added)
                                   ! Acknowledge
PC = PC + 2;
                                   ! Increment Program Counter
next;
                                   ! Execute Pending Assignments
T = 5:
next;
PHI1 = hi:
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 5
                                   ! Memory Read
RW = hi;
ADENABLE = 10;
                                   ! Disable Address Bus Buffer
LIBENABLE = 10;
                                   ! Disable Data Bus Buffer
DBUS = 0xffff;
                                   ! Data Bus High Impedanced
IARUS = A[1]:
                                   ! Place A[i] On Internal Address
                                   ! Rus
                                   ! Execute Pending Assignments
next;
PHI1 = 10;
                                   ! Phase 2 Of
PHI2 = hi;
                                   ! Clock Cycle 5
                                   ! Enable Address Bus Buffer
ADENABLE = hi;
                                   ! Gate Internal Address Bus
EXABUF = IABUS;
                                   ! Into External Address Buffer
                                   ! User Mode
FCMODE = SRMODE;
FCSPACE = 1;
                                   ! Accessing Program
                                   ! Execute Impending Assignments
next;
                                   ! Address Flaced On Bus(Added)
ABUS = EXABUF;
                                   ! Execute Pending Assignments
T = 6;
                                   ! Clock Cycle 6
next;
                                   ! Execute Assignment
                                   ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                                   ! Clock Cycle 6
ASN = 10;
                                   ! Assert Address Strobe
LIISN = lo:
                                   ! Assert Lower Data Strobe
DBENABLE = hi;
                                   ! Enable Data Bus
                                   ! Execute Pending Assignments
next;
PHI1 = lo;
                                   ! Phase 2
PHI2 = hi;
                                   ! Of Clock Cycle 6
next;
                                   ! Execute Pending Assignments
```

```
T = 7;
                             ! Clock Cycle 7
next;
                             ! Execute Assignment
PHI1 = hi;
                            ! Phase 1
                             ! Of Clock Cycle 7
PHI2 = 10;
while DTACKN eql hi
                            ! Wait For Memory To Place
                            ! Nata On The Bus
                            ! Execute Impending Assignments
    next;
    PH11 = 10;
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 7
                             ! Execute Assignments
    next;
    /************************
    T = 8;
                             ! Clock Cycle 8
    next;
                             ! Execute Assignment
                            ! Phase 1
    PHI1 = hi;
    PHI2 = 10;
                             ! Of Clock Cycle 8
    DRUS<7:0> = MEABUS3;
                            ! Place Byte On Data Bus And
    DTACKN = lo;
                             ! Asserts DTACKN(Added)
    next:
                             ! Execute Pending Assignments
    ! Return To Phase 2
                             ! Of Clock Cycle 7
    );
                          ! Execute Impending Assignments
    next;
T = 8;
                             ! Clock Cycle 8
                             ! Execute Assignment
next;
PHI1 = lo;
                             ! Phase 2
                             ! Of Clock Cycle 8
PHI2 = hi;
                             ! Instruction On Data Rus
EXDBUF = DBUS:
                             ! Is Placed In External Data
                             ! Bus Buffer
next;
                             ! Execute Pending Assignments
T = 9;
                             ! Clock Cycle 9
next:
                             ! Execute Assignment
PHI1 = h1;
                             ! Phase 1
PHI2 = 10;
                             ! Of Clock Cycle 9
                             ! Execute Pending Assignments
next;
PHI1 = 16;
                             ! Phase 2
                             ! Of Clock Cycle 9
PHI2 = hi;
HOLD = D[13<2:0> ext 4;
                             ! Place First 3 Bits Of D[1]
```

! Into Temporary Register

```
! The First 3 Bits Of D[1]
     case HOLD
          0: if EXDBUF<0>
                SRZERO = hi
                SRZERO = 10
          1: if EXDBUF<1>
                SKZERO = hi
             else
                SRZERO = lo
          2: if EXDBUF<2>
                                         ! Determine Which Bit Of
                SRZERO = hi
             else
                SRZER0 = 10
          3: if EXDBUF<3>
                SRZERO = hi
             else
                SRZERO = lo
          4: if EXDBUF<4>
                                        ! Data Will Be Checked For Set
                SRZERO = hi
             else
                SRZERO = 1o
          5: if EXDRUF<5>
                SRZERO = hi
             else
                SRZERO = 1o
          6: if EXDBUF<6>
                SRZERO = hi
             else
                SRZERO = 10
          7: if EXDBUF<7>
                SRZERO = hi
             else
                SRZERO = 10
     esac;
     ASN = hi;
                                          ! Deactivate Address Strobe
                                          ! Deactivate Lower Data Strobe
     LDSN = hi;
                                          ! Contents Of Prefetch Register
     IR = PFR;
                                          ! Are Placed Into Instruction
                                          ! Register
                                          ! Deactivate Data Transfer(Added)
     DTACKN = hi;
                                          ! Acknowledge
                                          ! Execute Pending Assignments
     next;
                                          ! Reset Clock Cycle Counter
     T = 0
                                          ! MOVE.W D2.D3
MOVE :=
     ! Phase 1 Of
     PHI1 = hi;
                                          ! Clock Cycle 0
     PHI2 = 10;
                                          ! Place Data Bus In High Impedance
     DBUS = 0xffff;
```

```
RW = hi:
                                   ! hemory Read
ADENABLE = 10;
                                   ! Disable Address Bus Buffer
DBENABLE = 10;
                                   ! Disable Data Rus Buffer
                                   ! Place PC On Internal Address
IABUS = PC;
IDBUS = D2LWORD;
                                   ! Place Low Word From D[2] Onto
                                    ! Internal Data Bus
next;
                                   ! Execute Pending Assignments
                                   ! Phase 2 Of
PHI1 = 10
PHI2 = hi:
                                   ! Clock Cycle O
ADENABLE = hi;
                                   ! Enable Address Bus Buffer
EXABUF = IABUS:
                                   ! Gate Internal Address Bus
                                   ! Into External Address Buffer
FCMODE = SRMODE;
                                   ! User Mode
FCSPACE = 2;
                                   ! Accessing Program
                                   ! Clear Status Register Carry Bit
SRCARRY = 10:
SROVER = lo;
                                   ! Clear Status Register Overflow Bit
SRZERO = 10;
                                    ! Clear Status Register Zero Bit
SRNEG = lo;
                                   ! Clear Status Register Negative Bit
D3LWORD = IDBUS;
                                   ! Place Data From Internal Data Bus
                                   ! Into Low Word Of D[3]
                                   ! Execute Impending Assignments
next;
ABUS = EXABUF;
                                   ! Address Placed On Rus(Added)
                                    ! Execute Pending Assignments
next;
T = 1;
                                    ! Clock Cycle 1
next:
                                    ! Execute Assignment
                                   ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                                   ! Clock Cycle 1
                                    ! Assert Address Strobe
ASN = lo;
                                    ! Assert Lower Data Strobe
LUSN = lo;
                                   ! Assert Upper Data Strobe
UDSN = 10;
                                   ! Enable Data Rus ·
DBENABLE = hi;
                                   ! Set Status Register Zero Bit
if D3LWORD eql 0
                                   ! If Moved Data Is Zero
   SRZERO = hi;
                                   ! Execute Pending Assignments
next;
PHI1 = 10;
                                   ! Phase 2
                                   ! Of Clock Cycle 1
PHI2 = hi:
if DC33<15>
                                   ! Set Status Register Negative
  SRNEG = hi;
                                  ! Bit If Moved Data Is Negative
                                   ! Execute Pending Assignments
next:
T = 2:
                                    ! Clock Cycle 2
next;
                                    ! Execute Assignment
PHI1 = hi;
                                    ! Phase 1
                                    ! Of Clock Cycle 2
PHI2 = 10;
                                   ! Wait For Memory To Place
while DTACKN eql hi
```

```
! Data On The Bus
     next;
                                ! Execute Impending Assignments
     PHI1 = lo;
                                ! Phase 2
     PHI2 = hi;
                                ! Of Clock Cycle 2
                                ! Execute Assignments
     next;
     T = 3;
                                ! Clock Cycle 3
     next;
                                ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
     PHI2 = 10:
                                ! Of Clock Cycle 3
     DBUS<15:8> = MCABUS3;
                                ! Memory Places Instruction
     DBUS<7:0> = MEARUS + 13;
                                ! On Data Rus And
     DTACKN = 10;
                                ! Asserts DTACKN(Added)
                                ! Execute Pending Assignments
     next;
     ! Return To Phase 2
                                ! Of Clock Cycle 2
     );
     next;
                                ! Execute Impending Assignments
! Clock Cycle 3
                                ! Execute Assignment
next;
                                ! Phase 2
PHI1 = 10;
                                ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                                ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Bus Buffer
next:
                                ! Execute Pending Assignments
! Clock Cycle 4
                                ! Execute Assignment
next;
PHI1 = hi;
                                ! Phase 1
PHI2 = 10;
                                ! Of Clock Cycle 4
                                ! The Contents Of The External
PFR = EXDBUF;
                                ! Data Bus Buffer Are Placed
                                ! In Prefetch Register
next;
                                ! Execute Pending Assignments
PHI1 = lo;
                                ! Phase 2
PHI2 = hi:
                                ! Of Clock Cycle 4
ASN = hi;
                                ! Deactivate Address Strobe
LDSN = hi;
                                ! Neactivate Lower Data Strobe
UDSN = hi;
                                ! Deactivate Upper Data Strobe
                                ! Contents Of Prefetch Register
IR = PFR;
                                ! Are Placed Into Instruction
```

```
! Register
     DTACKN = hi;
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
     PC = PC + 2;
                                         ! Increment Program Counter
     next;
                                        ! Execute Impending Assignments
     T = 0
                                         ! Reset Clock Cycle Counter
                                         ! JMP (A0)
jap :≃
     PHI1 = hi;
                                         ! Phase 1 Of
                                         ! Clock Cycle 0
     FHI2 = 10;
     DBUS = Oxffff;
                                        ! Place Data Bus In A High Impedance
     RW = hi;
                                         ! Memory Read
                                         ! Disable Address Bus Buffer
     ADENABLE = 10;
     DBENABLE = 10;
                                         ! Disable Data Bus Buffer
     IABUS = PC;
                                        ! Place PC On Internal Address
                                        ! Execute Pending Assignments
     next;
     PHI1 = 10;
                                        ! Phase 2 Of
     PHI2 = hi;
                                         ! Clock Cycle 0
                                         ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                        ! Gate Internal Address Bus
                                        ! Into External Address Buffer
                                        ! User Mode
     FCMODE = SRMODE;
     FCSPACE = 2;
                                        ! Accessing Program
                                        ! Execute Pending Assignments
     next:
                                        ! Address Flaced On Bus(Added)
     ARUS = EXABUF;
                                        ! Execute Pending Assignments
     next;
     T = 1;
                                         ! Clock Cycle 1
                                         ! Execute Assignment
     next;
                                        ! Phase 1 Of
     PHI1 = hi;
     FHI2 = 10;
                                         ! Clock Cycle 1
                                         ! Assert Address Strobe
     ASN = lo;
                                         ! Assert Lower Data Strobe
     LDSN = lo:
     UDSN = 10;
                                        ! Assert Upper Data Strobe
     IARUS = A[0];
                                         ! Move Jump Address From A[0]
                                         ! To Internal Address Buffer
     DBENABLE = hi;
                                         ! Enable Nata Bus
                                         ! Execute Pending Assignments
     next;
                                        ! Phase 2
     PHI1 = 10;
                                        ! Of Clock Cycle 1
     PH12 = hi;
```

! Place Jump Address Into Program

PC = IABUS;

```
! Counter
next;
T = 2;
                             ! Clock Cycle 2
                             ! Execute Assignment
next;
                             ! Phase 1
PHI1 = hi;
                             ! Of Clock Cycle 2
FHI2 = 10;
while DTACKN eql hi
                            ! Wait For Memory To Place
                            ! Data On The Bus
    next;
                            ! Execute Impending Assignments
    FH11 = 10;
                            ! Phase 2
                            ! Of Clock Cycle 2
    PHI2 = hi:
                             ! Execute Assignments
    next;
    ! Clock Cycle 3
    T = 3;
    next;
                             ! Execute Assignment
    PHI1 = hi;
                             ! Phase 1
    PHI2 = 10:
                             ! Of Clock Cycle 3
    INUS<15:8> = MEARUS3;
                             ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                            ! On Bata Bus And
    IITACKN = 10;
                             ! Asserts DTACKN(Added)
    next:
                             ! Execute Pending Assignments
    T = 2
                             ! Return To Phase 2
                             ! Of Clock Cycle 2
    );
    next;
                             ! Execute Impending Assignments
T = 3:
                             ! Clock Cycle 3
next;
                             ! Execute Assignment
PHI1 = lo;
                            ! Phase 2
                            ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                            ! Instruction On Data Bus
                            ! Is Placed In External Data
                             ! Bus Buffer
                             ! Execute Pending Assignments
next;
T = 4;
                             ! Clock Cycle 4
next;
                             ! Execute Assignment
PHI1 = hi;
                             ! Phase 1
PHI2 = 10;
                            ! Of Clock Cycle 4
next;
PFR = EXDBUF;
                            ! The Contents Of The External
```

```
! Data Bus Buffer Are Placed
                                  ! In Prefetch Register
next;
                                  ! Execute Pending Assignments
PHI1 = lo;
                                  ! Phase 2
                                  ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                                  ! Neactivate Address Strobe
LDSN = hi;
                                  ! Deactivate Lower Data Strobe
UIISN = hi;
                                 ! Deactivate Upper Data Strobe
DTACKN = hi;
                                  ! Deactivate Data Transfer
                                  ! Acknowledge(Added)
T = 5;
                                  ! Clock Cycle 5
next;
                                  ! Execute Previous Assignment
PHI1 = hi;
                                  ! Phase 1 Of
FHI2 = 10;
                                  ! Clock Cycle 5
RW = hi;
                                  ! Memory Read
ADENABLE = lo;
                                  ! Disable Address Bus Buffer
                                  ! Disable Data Bus Buffer
DRENABLE = 10;
                                  ! Place PC On Internal Address
IABUS = PC;
                                  ! Execute Pending Assignments
next;
PHI1 = lo;
                                 ! Phase 2 Of
PHI2 = hi;
                                  ! Clock Cycle 5
ADENABLE = hi;
                                  ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                 ! User Mode
                                 ! Accessing Program
FCSPACE = 2;
EXABUF = IABUS;
                                 ! Gate Internal Address Bus
next;
                                 ! Into External Address Buffer
ABUS = EXABUF;
                                  ! Address Flaced On Bus(Added)
next:
                                  ! Execute Pending Assignments
T = 6;
                                  ! Clock Cycle 6
                                  ! Execute Assignment
next;
                                  ! Phase 1 Of
PHI1 = hi;
                                  ! Clock Cycle 6
PHI2 = 10;
                                  ! Assert Address Strobe
ASN = lo:
LUSN = lo;
                                  ! Assert Lower Data Strobe
UDSN = lo;
                                 ! Assert Upper Data Strobe
DBENABLE = hi;
                                  ! Enable Data Rus
                                  ! Execute Pending Assignments
next;
F'HI1 = 10;
                                  ! Phase 2
PHI2 = hi;
                                  ! Of Clock Cycle 6
                                  ! Execute Pending Assignments
next;
! Clock Cycle 7
```

こうかんさん 連合ない ベドスト からく こうしゅ コフィックのかき マイイン さかかぬ フランシンション

```
! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi;
PHI2 = 10;
                               ! Of Clock Cycle 7
while DTACKN eql hi
                               ! Wait For Memory To Place
                               ! Data On The Bus
                               ! Execute Impending Assignments
     next;
     PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 7
     PHI2 = hi;
                             ! Execute Assignments
     next:
     ! Clock Cycle 8
     T = 8:
                                ! Execute Assignment
     next;
                               ! Phase 1
     PHI1 = hi;
                               ! Of Clock Cycle 8
     PHI2 = 10;
     DBUS<15:8> = MEABUS];
                               ! Memory Places Instruction
     DBUS<7:0> = MCABUS + 13;
                               ! On Data Bus And
                               ! Asserts DTACKN(Added)
     DTACKN = 10;
                                ! Execute Pending Assignments
     next;
     /*********************************
                               ! Return To Phase 2
     T = 7
                               ! Of Clock Cycle 7
     );
                               ! Execute Impending Assignments
     next;
T = 8:
                                ! Clock Cycle 8
                                ! Execute Assignment
next;
                                ! Phase 2
PHI1 = lo;
                                ! Of Clock Cycle 8
PHI2 = hi;
EXDBUF = DRUS;
                                ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Bus Buffer
                                ! Execute Pending Assignments
next;
! Clock Cycle 9
T = 9;
                                ! Execute Assignment
next;
                                ! Phase 1
PHI1 = hi;
                                ! Of Clock Cycle 9
PHI2 = 16:
                                ! The Contents Of The External
PFR = EXD&UF;
                                ! Data Bus Buffer Are Placed
                                ! In Prefetch Register
                                ! Execute Pending Assignments
next;
                                ! Phase 2
PHI1 = lo:
                                ! Of Clock Cycle 9
PHI2 = hi;
```

ストラントラ 日本大人の大人の

```
ASN = hi;
                                             ! Deactivate Address Strobe
     LDSN = hi;
                                             ! Deactivate Lower Data Strobe
     UDSN = hi;
                                             ! Deactivate Upper Data Strobe
                                             ! Increment Program Counter
     PC = PC + 2;
                                             ! Place Contents Of Prefetch
     IR = PFR;
                                             ! Register Into Instruction
                                             ! Register
      DTACKN = hi;
                                             ! Deactivate Data Transfer
                                             ! Acknowledge(Added)
                                             ! Execute Pending Assignments
      next;
      T = 0
                                             ! Reset Clock Cycle Counter
      )
decode_execute_prefetch :=
                        case IR
                                             ! BTST D1,(A1)
                             0x0311: btst
                                             ! MOVE.W D2,D3
                             0x3602: move
                                             ! JMP (AO) If IR = Octal Value
                             Ox4edO: jmp
                        esac
                        )
main :=
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
           decode_execute_prefetch
```

)

```
/*
                                                */
/*
     MOTOROLA MC68000 MODEL OF THE ILLEGAL INSTRUCTION
                                                */
/*
                                                */
/*
                                                */
/*
              Structure Declarations
                                                */
/*
                                                */
state
/*
                                                */
/*
           M68000 Programming Registers
                                                */
. /*
                                                x/
DE0:73<31:0>,
                   ! 8 Nata Registers
AE0:63<31:0>,
                   ! 7 Address Registers
UA7<31:0>,
                   ! User Stack Pointer
SA7<31:0>,
                   ! System Stack Pointer
PC<31:0>,
                   ! Program Counter
SR<15:0>,
                   ! Status Register
/¥
                                                */
/*
                                                */
            Temporary Internal Registers
/*
                                                x/
PFR<15:0>,
                   ! Prefetch Register
IR<15:0>.
                   ! Instruction Register
FC<2:0>,
                   ! Function Code Register
EXDBUF<15:0>,
                   ! External Data Bus Buffer Register
EXABUF<23:1>,
                   ! External Address Bus Buffer Register(changed)
                   ! ALU Buffer 1
ALUBUF1<31:0>,
ALUBUF2<31:0>,
                   ! ALU Buffer 2
DTEMP<15:0>,
                   ! Temporary Data Storage
                   ! Temporary Displacement Storage
DISREG<31:0>,
SRTEMP<15:0>,
                   ! Temporary Status Register Storage
                   ! (Exception Processing)
IRTEMP<15:0>,
                   ! Temporary Instruction Register Storage
                   ! (Exception Processing)
TEMPADR<31:0>,
                   ! Temporary Cycle Address Storage
                   ! (Exception Processing)
ACTYPE<15:0>,
                   ! Temporary Access Type Storage
                   ! (Exception Processing)
VECADR<23:0>.
                   ! Temporary Vector Address Storage
                   ! (Exception Processing)
```

```
HANADR<31:0>,
                         ! Temporary Address Storage For
                         ! Exception Handler Routine
T<7:0>,
                         ! Clock Cycle Counter
RESET.
                       ! Reset Flip-Flop
HALT,
                      ! Halt Flip-Flop
RW,
                      ! Read/Write Flip-Flop
ADENABLE,
                      ! Address Bus Buffer Enable
DIBENABLE,
                      ! Data Bus Buffer Enable
ASN.
                       ! Address Strobe Flip-Flop
LDSN.
                       ! Lower Nata Strobe Flip-Flop
UDSN.
                       ! Upper Data Strobe Flip-Flop
DITACKN.
                       ! Data Transfer Acknowledge Flip-Flop
COUT,
                       ! Carry Flip-Flop
EXCEPT,
                      ! Exception Processing Flip-Flop
REALLY,
                       ! Ready Flip-Flop
/*
                                                                 */
/*
       Model transformation modifications:
                                                                 */
/*
                                                                 */
/*
           1) CDL decoder structure nonexistent in ISP' and un-
                                                                 */
/*
       necessary for model. Eliminated.
                                                                 */
/*
           2) Multi-phase clock structure nonexistent in ISP'.
                                                                 */
/*
       Operations on registers will provide its equivalent.
                                                                 */
           3) Switch structure nonexistent in ISP'. Operation on a
/*
                                                                 */
/*
       register will provide its equivalent.
                                                                 x/
/*
           4) The declared bus structures are modeled with registers
/*
       without loss of model accurracy. This done to maintain model
                                                                 */
14
       equivalency and simplicity.
                                                                 */
/*
           5) The memory word length was reduced from 16 to 8 bit
/*
       words to coincide with the ECR's 32-Kbyte memory, to agree with*/
/*
       their FC incrementation, and to enable the use of existing
                                                                 */
       MC68000 assembler and linker/loader models. The memory was
/x
                                                                 */
/*
       also reduced from 8 Mwords to 32 Kbytes.
                                                                 */
/*
                                                                 */
TABUS<31:0>,
                         ! Internal Address Bus
IDBUS<31:0>,
                         ! Internal Data Bus
twait<7:0>,
                       ! Power Switch
SWITCH.
                       ! Phase 1 Of Two-Phase Clock
PHI1.
                       ! Phase 2 Of Two-Phase Clock
FHI2;
port
/*
                                                                 */
/*
              External Address and Data Bus
                                                                 */
/*
DBUS<15:0>,
                         ! External Data Bus
```

AND SOUNDS INCOME TO SOUND IN THE SOUND IN T

```
ABUS<23:1>;
                      ! External Address Bus(changed)
format
/*
/*
                 Register Subfields
                                                         */
/*
                                                         */
PCADDR
         = PC<2310>,
                      ! Program Counter Address Field
SRTRACE
                      ! Trace Bit
         = SR\langle 15\rangle,
         = SR<13>,
                      ! Mode Selection Bit
SRMODE
         = SR<0>,
SRCARRY
                      ! Carry Bit
SROVER
                      ! Overflow Bit
         = SR<1>,
         = SR<2>,
                      ! Zero Bit
SRZERO
                      ! Negative Bit
SKNEG
         = SR<3>,
SREX
         = SR<4>,
                      ! Extend Bit
SRMASK
         = SR<10:8>,
                      ! Interrupt Mask
FCSFACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCMODE
         = FC<2>,
                      ! User/Supervisor Mode Bit
F'CLOW
         = PC<15:0>.
                      ! PC Low Word
         = PC<31:16>,
PCHI
                      ! PC High Word
I(OLWORI)
         = DE03<15:0>,
                      ! DEOJ Low Word
DILWORD
         = DE1J<15:0>,
                      ! D[1] Low Word
D2LWORD
         = DE23<15:0>,
                      ! DE23 Low Word
                      ! DE33 Low Word
D3LWORD
         = D[3]<15:0>,
DALWORD
         = B[4](15:0),
                      ! D[4] Low Word
DSLWORD
         = DC53<15:0>,
                      ! DES] Low Word
DI6LWORD
         = DE63<15:0>,
                      ! DC63 Low Word
D7LWORD
         = DE7J<15:0>,
                      ! D[7] Low Word
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANATIRLOW
         = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPADRHI
         = TEMPADR<31:16>;! TEMPADR High Word
MEMORY
/*
                                                         x/
/*
                16K 16-Bit Word Internal Memory
                                                         */
1%
                                                         */
MEO:327673<7:0>;
BIGCTO
/*
                                                         1/
/*
                Logic Level Macros
                                                         */
```

```
/x
10
     = 0 2,
hi
     = 1 &.
off
     = 0 %,
     = 1 %.
clear = 0 %:
/*
/* Fower On and Initialization. This process was not modeled but is
                                                            */
   added to initialize signals and registers.
                                                            */
/*
                                                            */
power_on_initialize :=
      SWITCH = on;
                                   ! Turn Power On
                                   ! Execute Assignment
       next;
      READY = lo;
                                   ! System Not Ready
      RESET = 10;
                                   ! Assert Reset For
       delay(100);
                                   ! 100 Miliseconds(Active Low)
      RESET = hi;
                                   ! Deactivate Reset
       next;
                                   ! Execute Pending Assignments
       ASN = hi;
                                   ! Initialize Address Strobe
      LISN = hi:
                                   ! Initialize Lower Data Strobe
      UDSN = hi;
                                   ! Initialize Upper Data Strobe
      DITACKN = hi;
                                   ! Initialize Bata Transfer Acknowledge
      RW = hi;
                                   ! Initialize Read/Write(Read On High)
      DBUS = Oxffff;
                                  ! Place Data Bus in High Impedance State
      ABUS = 0xffffff;
                               ! Place Address Rus In High Impedance State
      M[0x100c] = 0xff;
                                   ! Place Memory Locations Following The
      MEO \times 100dJ = O \times ff;
                                   ! JMP Instruction In A High State
      HALT = hi;
                                   ! Initialize Halt Flip-Flop(Active
                                   ! Low)
       T = 0;
                                   ! Initialize Clock Cycle Counter
       READY = hi;
                                   ! System Ready
       /*
                                                            */
      /*
            Routine Initialization Fer Hampy and Guillory
                                                            */
       /*
                                                            */
       SRMODE = 10;
                                   ! Set Status Register To User Mode
       D[1] = 0 \times 0;
                                   ! Place Hex O Into D[1]
                                   ! Place Hex 1000 Into A[0]
       A[0] = 0 \times 1004;
       PC = 0 \times 1004;
                                   ! Place Hex 1000 Into Program Counter
       ME0\times2002] = 0\times4e;
                                   ! Place RTE Instruction In
       M[0x2003] = 0x73;
                                   ! Location 2000 Hex
       M[0x10] = 0x0;
                                   ! Place Exception
       M[0x11] = 0x0;
                                   ! Vector In
      M[0x12] = 0x20;
                                   ! Location
       ME0x133 = 0x02;
                                   ! 10 Hex
```

とという。「日本・ハントントン」というのではない。「日本のでは、「日本のでは、日本のではのでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、日本のでは、

```
SA7 = 0x08e6:
                                  ! Initialize System Stack Pointer
                                  ! At Program Headed Down (Added)
                                  ! Execute Assignments
      next
/* Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                           */
/* to retrieve modeled instructions for simulation and analysis. It
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
/# on page VI-15 of their thesis.
                                                           */
fetch_initial_instruction :=
     PHI1 = hi:
                                    ! Phase 1 Of
    PHI2 = 16;
                                    ! Clock Cycle 0
    RW = hi;
                                    ! Memory Read
                                    ! Disable Address Bus Buffer
     ADENABLE = 10;
                                    ! Disable Data Bus Buffer
    DRENABLE = lo;
     IABUS = PC;
                                    ! Place PC On Internal Address
                                    ! Bus
    next;
                                    ! Execute Pending Assignments
    PHI1 = 10;
                                    ! Phase 2 Of
    PHI2 = hi;
                                    ! Clock Cycle 0
                                    ! Enable Address Bus Buffer
    ADENABLE = hi;
    EXABUF = IABUS;
                                    ! Gate Internal Address Rus
                                    ! Into External Address Buffer
     FCMODE = SRMODE;
                                    ! User Mode
    FCSPACE = 2;
                                    ! Accessing Program
     next;
                                    ! Execute Impending Assignments
     ABUS = EXABUF;
                                    ! Address Placed On Bus(Added)
                                    ! Execute Pending Assignments
     next;
     T = 1:
                                    ! Clock Cycle 1
     next;
                                    ! Execute Assignment
    PHI1 = hi;
                                    ! Phase 1 Of
     PHI2 = 10;
                                    ! Clock Cycle 1
     ASN = lo;
                                    ! Assert Address Strobe
    LDSN = 10;
                                    ! Assert Lower Data Strobe
     UDSN = lo;
                                    ! Assert Upper Data Strobe
     DBENABLE = hi:
                                    ! Enable Data Rus
     next;
                                    ! Execute Pending Assignments
    PHI1 = 10;
                                    ! Phase 2
```

! Of Clock Cycle 1

PHI2 = hi;

```
next;
                             ! Execute Fending Assignments
T = 2;
                             ! Clock Cycle 2
next;
                             ! Execute Assignment
                             ! Phase 1
PHI1 = hi;
                            ! Of Clock Cycle 2
PHI2 = 10;
while DTACKN eql hi
                            ! Wait For Memory To Place
                            ! Data On The Bus
    next;
                            ! Execute Impending Assignments
    FHI1 = lo;
                            ! Phase 2
    PHI2 = hi;
                            ! Uf Clock Cycle 2
                            ! Execute Assignments
    next;
    T = 3;
                            ! Clock Cycle 3
    next;
                             ! Execute Assignment
    PHI1 = hi;
                            ! Phase 1
    FHI2 = 10;
                            ! Of Clock Cycle 3
                            ! Memory Places Instruction
    DBUS<15:8> = MCABUSJ;
    DBUS<7:0> = MEABUS + 13;
                            ! On Data Bus And
    DTACKN = 10;
                            ! Asserts DTACKN(Added)
    next;
                             ! Execute Pending Assignments
    ! Return To Phase 2
                            ! Of Clock Cycle 2
    );
                            ! Execute Impending Assignments
    next;
! Clock Cycle 3
T = 3;
next;
                             ! Execute Assignment
PHI1 = 10;
                            ! Pflase 2
                            ! Of Clock Cycle 3
PHI2 = hi;
EXDBUF = DBUS;
                            ! Instruction On Data Rus
                            ! Is Placed In External Data
                             ! Bus Ruffer
                             ! Execute Pending Assignments
next;
T = 4;
                            ! Clock Cycle 4
                             ! Execute Assignment
next;
                            ! Phase 1
PHI1 = hi;
                            ! Of Clock Cycle 4
PHI2 = 10;
FFR = EXDBUF;
                            ! The Contents Of The External
                             ! Data Bus Buffer Are Placed
                             ! In Prefetch Register
```

```
! Execute Pending Assignments
     next;
                                        ! Phase 2
     PHI1 = lo;
     PHI2 = hi;
                                        ! Of Clock Cycle 4
     ASN = hi:
                                        ! Deactivate Address Strobe
     LDSN = hi;
                                        ! Deactivate Lower Data Strobe
                                        ! Deactivate Upper Data Strobe
     UDSN = hi;
                                        ! Contents Of Prefetch Register
     IR = PFR:
                                        ! Are Placed Into Instruction
                                        ! Register
     DTACKN = hi;
                                        ! Deactivate Data Transfer(Added)
                                        ! Acknowledge
     PC = PC + 2;
                                        ! Increment Program Counter
     next;
                                        ! Execute Pending Assignments
     T = 0
                                        ! Reset Clock Cycle Counter
     )
illegal :=
                                        ! Illegal Instruction (4AFC)
       PHI1 = hi;
                                        ! Phase 1 Of
     PHI2 = 16;
                                        ! Clock Cycle 0
                                        ! Memory Read
     RW = hi;
     ADENABLE = 10;
                                        ! Disable Address Bus Buffer
                                       ! Address Bus High Impedanced
     ABUS = 0xffffff;
     DRENABLE = 10:
                                       ! Disable Data Bus Duffer
                                       ! Data Bus High Impedanced
     DBUS = 0xffff;
     IABUS<31:1> = PC<31:1>;
                                        ! Place FC On Internal Address
                                        ! Bus
                                        ! Execute Pending Assignments
     next;
     PHI1 = 10;
                                        ! Phase 2 Of
     PHI2 = hi;
                                        ! Clock Cycle 0
                                       ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS<23:1>;
                                       ! Gate Internal Address Bus
                                       ! Into External Address Buffer
     FCMODE = SRMUDE;
                                        ! User Mode
     FCSPACE = 2;
                                        ! Accessing Program
     ABUS = IABUS<23:1>;
                                        ! Address Placed On Bus(Added)
                                        ! Execute Fending Assignments
     next;
     T = 1;
                                        ! Clock Cycle 1
     next;
                                        ! Execute Assignment
                                        ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                        ! Clock Cycle 1
     ASN = lo;
                                        ! Assert Address Strobe
     LDSN = 10;
                                        ! Assert Lower Data Strobe
     UDSN = lo:
                                        ! Assert Upper Data Strobe
```

DBENABLE = hi;

! Enable Data Bus

```
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                              ! Phase 2
PHI2 = hi;
                              ! Of Clock Cycle 1
IDBUS = SR;
                              ! Place Status Register On
                              ! Internal Data Bus
PC = PC - 2;
                              ! Decrement PC To Illegal Instruction
                              ! Address (Changed)
                              ! Execute Pending Assignments
next;
T = 2;
                               ! Clock Cycle 2
next;
                               ! Execute Assignment
PHI1 = hi;
                              ! Phase 1
                              ! Of Clock Cycle 2
PHI2 = 10;
                              ! Place Status Register
SRITEMP = IDBUS;
                              ! On Bus In Temporary Register
while DTACKN eql hi
                              ! Wait For Memory To Place
                              ! Data On The Bus
                              ! Execute Impending Assignments
    next;
    PHI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 2
    PHI2 = hi;
    next;
                              ! Execute Assignments
     ! Clock Cycle 3
     T = 3;
    next;
                              ! Execute Assignment
     PH11 = hi;
                              ! Phase 1
    PHI2 = 10;
                              ! Of Clock Cycle 3
     DBUS<15:8> = MCABUS];
                              ! Memory Places Instruction
                             ! On Data Bus And
     [BUS<7:0> = M[ABUS + 1];
    IITACKN = lo;
                              ! Asserts DTACKN(Added)
                               ! Execute Pending Assignments
     next:
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
     ):
                              ! Execute Impending Assignments
     next;
T = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
PHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DRUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
```

! Set Supervisor State

SRMODE = hi;

```
SRTRACE = lo;
                                 ! Turn Off Trace
                                 ! Execute Pending Assignments
next;
T = 4;
                                 ! Clock Cycle 4
                                 ! Execute Assignment
next;
PHI1 = hi;
                                 ! Phase 1
                                 ! Of Clock Cycle 4
PHI2 = lo;
                                 ! The Contents Of The External
PFR = EXDBUF;
                                 ! Data Bus Buffer Are Placed
                                 ! In Prefetch Register
SA7 = SA7 - 2;
                                 ! Decrement System Stack Fointer
                                 ! To Point To Location That Will
                                 ! Receive PC's Low Word
                                 ! Execute Pending Assignments
next;
FHI1 = 10;
                                 ! Phase 2
                                 ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                                 ! Deactivate Address Strobe
                                 ! Deactivate Lower Data Strobe
LDSN = hi;
                                 ! Deactivate Upper Data Strobe
UDSN = hi;
DTACKN = hi;
                                 ! Deactivate Data Transfer(Added)
                                 ! Acknowledge
next;
                                 ! Execute Pending Assignments
! Clock Cycle 5
T = 5:
next;
PHI1 = hi;
                                 ! Phase 1 Of
                                 ! Clock Cycle 5
PHI2 = 10;
ADENABLE = lo;
                                ! Disable Address Bus Buffer
ABUS = 0xffffff;
                                ! Address Bus High Impedanced
LIBENABLE = 10;
                                 ! Disable Data Bus Buffer
DBUS = 0xffff;
                                 ! Data Bus High Impedanced
VECAUR = 4;
                                 ! Place Vector Number In Register
next:
                                 ! Execute Pending Assignments
                                 ! Phase 2 Of
PHI1 = lo:
PH12 = hi;
                                 ! Clock Cycle 5
                                 ! Execute Impending Assignments
next;
T = 6;
                                 ! Clock Cycle 6
                                 ! Execute Assignment
next;
                                 ! Phase 1 Of
PHI1 = hi;
                                 ! Clock Cycle 6
PHI2 = 10;
VECAUR = VECAUR *: arith 2;
                                ! Multiply Vector Number
                                 ! By 4 For Vector Address
                                 ! Execute Pending Assignments
next;
```

```
PHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 6
next;
                                ! Execute Pending Assignments
T = 7;
                                ! Clock Cycle 7
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1
FHI2 = 10;
                                ! Of Clock Cycle 7
next:
                                ! Phase 2
PHI1 = lo:
PHI2 = hi;
                                ! Of Clock Cycle 7
T = 8:
                                ! Clock Cycle 8
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1
FH12 = 10;
                                ! Of Clock Cycle 8
                                ! Execute Pending Assignments
next;
FHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 8
next;
! Clock Cycle 9
T = 9:
next;
                                 ! Execute Assignment
PHI1 = hi;
                                ! Phase 1 Of
PHI2 = 10;
                                ! Clock Cycle 9
RW = hi;
                                ! Memory Read
                                ! Disable Address Bus Buffer
ADENABLE = 10;
                                ! Disable Data Bus Buffer
DIBENABLE = 10;
                                ! Data Bus High Impedanced
DBUS = Oxffff;
IARUS = SA7;
                                 ! Place SA7 On Internal Address
                                 ! Rus
next;
                                 ! Execute Pending Assignments
PHI1 = 10;
                                ! Phase 2 Of
                                ! Clock Cycle 9
PHI2 = hi;
                                ! Enable Address Bus Buffer
ADENABLE = hi;
EXABUF = IABUS<23:1>;
                                ! Gate Internal Address Bus
                                ! Into External Address Buffer
                                ! Supervisor Mode
FCMODE = SRMODE;
FCSFACE = 1;
                                ! Accessing Data
IDBUS = PCLOW:
                                ! Place Low Word from PC Onto
                                 ! Internal Data Rus
ABUS = IABUS<23:1>;
                                 ! Place Address On External Data Bus
```

ステストラング 自動の たんかかん 安備 でなる ないない (機能の)

```
next;
                                 ! Execute Impending Assignments
T = 10;
                                 ! Clock Cycle 10
next:
                                 ! Execute Assignment
                                 ! Phase 1 Of
PHI1 = hi:
PHI2 = 10;
                                 ! Clock Cycle 10
ASN = 10;
                                 ! Assert Address Strobe
£₩ = 10;
                                 ! Activate Write
EXDBUF = IDBUS;
                                 ! Place Internal Data Bus
                                 ! Contents Into External Data Buffer
next;
                                 ! Execute Pending Assignments
                                 ! Phase 2
PHI1 = 10;
PHI2 = hi;
                                 ! Of Clock Cycle 10
DRUS = EXDBUF;
                                 ! Contents Of External Data Buffer
                                ! Placed On Data Bus
DBENABLE = hi;
                                ! Enable Data Bus
next;
                                ! Execute Pending Assignments
T = 11;
                                 ! Clock Cycle 11
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
PHI2 = lo:
                                 ! Of Clock Cycle 11
UDSN = lo;
                                 ! Activate Upper Data Strobe
LUSN = 10;
                                  ! Activate Lower Data Strobe
twait = 0;
next;
while DTACKN eql hi
                                 ! Wait For Memory To Place
                                 ! Data On The Bus
     twait = twait + 1;
     next;
                                 ! Execute Impending Assignments
     PHI1 = 10;
                                 ! Phase 2
     PHI2 = hi;
                                 ! Of Clock Cycle 11
     next:
                                 ! Execute Assignments
     T = 12;
                                 ! Clock Cycle 12
     next:
                                 ! Execute Assignment
     PHI1 = hi:
                                 ! Phase 1
     PH12 = 10;
                                 ! Of Clock Cycle 12
     if twait eql 2
     M[ABUS] = DBUS<15:8>;
                                ! PC Low Word
     MEARUS + 13 = DBUS<7:0>;
                                 ! Stored
                                ! Asserts DTACKN(Added)
     ITACKN = 10
     );
     next;
                                ! Execute Pending Assignments
```

```
T = 11
                                ! Return To Phase 2
                               ! Of Clock Cycle 11
     );
                               ! Execute Impending Assignments
     next;
T = 12:
                                ! Clock Cycle 12
next;
                                ! Execute Assignment
PHI1 = lo;
                                ! Phase 2
PH12 = hi;
                               ! Of Clock Cycle 12
SA7 = SA7 - 4;
                               ! Set System Stack Pointer
                               ! To Point To Status Register
                                ! Storage Location
next;
                               ! Execute Pending Assignments
T = 13;
                                ! Clock Cycle 13
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 13
PHI2 = 10;
next;
                                ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
                                ! Of Clock Cycle 13
PHI2 = hi;
ASN = hi;
                                ! Deactivate Address Strobe
LUSN = hi;
                                ! Deactivate Lower Data Strobe
UDSN = hi;
                                ! Deactivate Upper Data Strobe
DTACKN = hi;
                                ! Deactivate Data Transfer(Added)
                                ! Acknowledge
next;
                                ! Execute Pending Assignments
T = 14;
                                ! Clock Cycle 14
next;
                                ! Phase 1 Of
PHI1 = hi;
                                ! Clock Cycle 14
PHI2 = 1o;
RW = hi;
                                ! Memory Read
                                ! Disable Address Bus Buffer
ADENABLE = 10;
                                ! Address Bus High Impedanced
ABUS = 0xffffff;
DBENABLE = 10;
                                ! Disable Data Bus Buffer
                                ! Place SA7 On Internal Address
IABUS = SA7:
                                ! Rus
                                ! Execute Pending Assignments
next;
                                ! Phase 2 Of
PHI1 = 10;
                                ! Clock Cycle 14
PHI2 = hi;
ADENABLE = hi;
                                ! Enable Address Bus Buffer
```

ことには自ちさきたからない。自身などになってい

! Nata Rus High Impedanced

DBUS = Oxffff;

```
EXABUF = IABUS<23:1>;
                                ! Gate Internal Address Bus
                                ! Into External Address Buffer
FCMODE = SRMODE;
                                ! Supervisor Mode
FCSFACE = 1;
                                ! Accessing Data
IDBUS = SRTEMP;
                                ! Place Holder Of Status Register
                                ! Onto Internal Data Bus
ABUS = IABUS<23:1>;
                                ! Place Address On Address Bus
                                ! Execute Impending Assignments
next;
! Clock Cycle 15
                                 ! Execute Assignment
next;
                                 ! Phase 1 Of
PHI1 = hi:
                                 ! Clock Cycle 15
PHI2 = 10;
                                 ! Assert Address Strobe
ASN = 10:
RW = 10:
                                 ! Activate Write
EXDRUF = IDBUS;
                                 ! Internal Data Bus Moved
                                 ! To External Data Buffer
next;
                                 ! Execute Pending Assignments
PHI1 = 10;
                                ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 15
                                 ! Contents Of External Data
DBUS = EXDBUF:
                                 ! Buffer Placed On Data Bus
DBENABLE = hi;
                                 ! Enable Data Bus
                                 ! Execute Pending Assignments
next:
T = 16;
                                 ! Clock Cycle 16
next;
                                 ! Execute Assignment
FHI1 = hi;
                                 ! Phase 1
PHI2 = 10;
                                 ! Of Clock Cycle 16
                                 ! Activate Upper Nata Strobe
UNSN = lo:
LDSN = 10;
                                 ! Activate Lower Data Strobe
twait = 0:
next:
                              ! Wait For Memory To Place
while DTACKN eql hi
                                 ! Data On The Bus
     twait = twait + 1;
     next;
                                ! Execute Impending Assignments
     PHI1 = lo;
                                 ! Phase 2
     PH12 = hi;
                                 ! Of Clock Cycle 16
                                 ! Execute Assignments
     next;
     T = 17;
                                 ! Clock Cycle 17
     next;
                                ! Execute Assignment
                                 ! Phase 1
     PHI1 = hi;
                                 ! Of Clock Cycle 17
     PHI2 = 10;
```

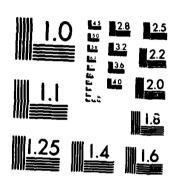
(

```
if twait eq1 2
                              ! PC Low Word
     MCARUS] = DBUS<15:8>;
     MCABUS + 13 = DBUS<7:0>;
                               ! Stored
                              ! Asserts DTACKN(Added)
     ITACKN = 10
     ):
                              ! Execute Fending Assignments
     next:
     ! Return To Phase 2
                               ! Of Clock Cycle 16
     );
     next;
                               ! Execute Impending Assignments
T = 17;
                               ! Clock Cycle 17
                               ! Execute Assignment
next;
FHII = lo;
                               ! Phase 2
                               ! Of Clock Cycle 17
PHI2 = hi;
SA7 = SA7 + 2;
                              ! Set System Stack Pointer
                              ! To Point To High FC
                               ! Storage Location
                               ! Execute Pending Assignments
next;
T = 18;
                               ! Clock Cycle 18
                               ! Execute Assignment
next:
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 18
next;
                               ! Execute Pending Assignments
                               ! Phase 2
PHI1 = 10;
                               ! Of Clock Cycle 18
PHI2 = hi;
                               ! Deactivate Address Strobe
ASN = hi:
LDSN = hi;
                               ! Deactivate Lower Data Strobe
UIISN = hi;
                               ! Deactivate Upper Data Strobe
                               ! Deactivate Data Transfer(Added)
DTACKN = hi;
                               ! Acknowledge
next;
                               ! Execute Pending Assignments
! Clock Cycle 19
T = 19;
next;
                               ! Execute Pending Assignments
PHI1 = hi;
                               ! Phase 1 Of
                               ! Clock Cycle 19
PHI2 = 10;
                               ! Memory Read
RW = hi;
                               ! Disable Address Bus Buffer
ADENABLE = 10;
ABUS = OXFFFFFFF
                               ! Address Bus High Impedanced
INTENABLE = 10;
                               ! Disable Data Bus Ruffer
IABUS = SA7;
                               ! Place SA7 On Internal Address
```

```
! Bus
next;
                                  ! Execute Pending Assignments
FHI1 = lo;
                                 ! Phase 2 Of
PHI2 = hi:
                                 ! Clock Cycle 19
ADENABLE = hi;
                                 ! Enable Address Bus Buffer
DBUS = 0xffff;
                                  ! Data Bus High Impedanced
EXABUF = IABUS<23:1>;
                                 ! Gate Internal Address Bus
                                 ! Into External Address Ruffer
FCMODE = SRMODE;
                                 ! Supervisor Mode
FCSPACE = 1;
                                 ! Accessing Data
IDBUS = PCHI;
                                  ! Place High Word Of FC
                                  ! Onto Internal Data Bus
ABUS = IABUS<23:1>;
                                  ! Place Address On External Bus
                                 ! Execute lapending Assignments
next:
T = 20:
                                  ! Clock Cycle 20
next:
                                  ! Execute Assignment
                                  ! Phase 1 Of
PHI1 = hi;
PHI2 = 10:
                                 ! Clock Cycle 20
ASN = 10;
                                 ! Assert Address Strobe
RW = lo:
                                 ! Activate Write
EXDBUF = IDBUS;
                                 ! Internal Data Bus Moved
                                  ! To External Data Buffer
                                  ! Execute Pending Assignments
next;
                                 ! Phase 2
PHI1 = 10;
FHI2 = hi;
                                 ! Of Clock Cycle 20
DBUS = EXDBUF;
                                 ! Contents Of External Data
                                 ! Buffer Placed On Nata Bus
DBENABLE = hi;
                                  ! Enable Data Rus
next:
                                  ! Execute Pending Assignments
1 = 21:
                                  ! Clock Cycle 21
next;
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1
PHI2 = 10;
                                  ! Of Clock Cycle 21
                                 ! Activate Upper Data Strobe
UDSN = lo;
                                  ! Activate Lower Data Strobe
LDSN = lo;
twait = 0;
next;
while DTACKN eql hi
                                  ! Wait For Memory To Place
                                  ! Data On The Bus
     twait = twait + 1;
     next;
                                  ! Execute Impending Assignments
     PHI1 = lo;
                                  ! Phase 2
     FHI2 = hi;
                                  ! Of Clock Cycle 21
     next;
                                  ! Execute Assignments
```

```
T = 22;
                             ! Clock Cycle 22
    next;
                             ! Execute Assignment
    PHI1 = hi;
                            ! Phase 1
    PHI2 = 10:
                             ! Of Clock Cycle 22
    if twait eql 2
                           ! FC Low Word
    MEABUSD = DBUS<15:8>:
    MEABUS + 1] = DBUS<7:0>;
                            ! Stored
    LITACKN = 10
                            ! Asserts DTACKN(Added)
    );
    next;
                            ! Execute Pending Assignments
    ! Return To Phase 2
    T = 21
                            ! Of Clock Cycle 21
    );
                            ! Execute Impending Assignments
    next;
! Clock Cycle 22
T = 22;
next;
                             ! Execute Assignment
PHI1 = lo;
                             ! Phase 2
                            ! Of Clock Cycle 22
PHI2 = hi:
next:
                            ! Execute Pending Assignments
! Clock Cycle 23
T = 23:
next;
                            ! Execute Assignment
PHI1 = hi;
                            ! Phase 1
                            ! Of Clock Cycle 23
PHI2 = 10;
next;
                             ! Execute Pending Assignments
                            ! Phase 2
PHI1 = lo;
                            ! Of Clock Cycle 23
PHI2 = hi;
                            ! Deactivate Address Strobe
ASN = hi;
                            ! Deactivate Lower Data Strobe
LUSN = hi;
UDSN = hi:
                             ! Deactivate Upper Data Strobe
DITACKN = hi;
                            ! Deactivate Data Transfer(Added)
                            ! Acknowledge
                             ! Execute Fending Assignments
next;
T = 24:
                              ! Clock Cycle 24
                             ! Execute Assignment
nexti
PHI1 = hi;
                            ! Phase 1
PHI2 = 10;
                             ! Of Clock Cycle 24
```

THE SIMULATION AND ANALYSIS OF A RTL MODEL OF THE MOTOROLA MCS8000 MICROP. (U) AIR FORCE INST OF TECH MRIGHT-PATTERSON AFB OH SCHOOL OF ENGI. C A BAXLEY DEC 84 AFIT/GCS/ENG/840-2-VOL-2 F/G 9/2 MD-R164 257 5/5 UNCLASSIFIED FRIMED



MICROCOPY RESOLUTION TEST CHART

```
! Memory Read
RW = hi;
ADENABLE = lo;
                                  ! Disable Address Bus Buffer
                                  ! Address Bus High Impedanced
ABUS = 0xffffff;
DIRENABLE = 10;
                                  ! Disable Data Bus Buffer
1ABUS = VECABR;
                                  ! Place VECADA On Internal Address
                                  ! Execute Fending Assignments
next;
                                  ! Phase 2 Of
PHI1 = 10;
PHI2 = hi;
                                  ! Clock Cycle 24
ADENABLE = hi;
                                  ! Enable Address Bus Buffer
DRUS = 0xffff;
                                 ! Data Rus High Impedanced
EXABUF = IABUS<23:1>;
                                 ! Gate Internal Address Bus
                                  ! Into External Address Ruffer
FCMODE = SRMODE;
                                  ! Supervisor Mode
FCSPACE = 1;
                                  ! Accessing Data
ARUS = IABUS<23:1>;
                                  ! Place Address On External Bus
next;
                                  ! Execute Impending Assignments
\***********************************
T = 25;
                                  ! Clock Cycle 25
                                  ! Execute Assignment
next;
PHI1 = hi;
                                  ! Phase 1 Of
                                  ! Clock Cycle 25
PHI2 = lo;
ASN = 10;
                                  ! Assert Address Strobe
                                  ! Assert Lower Data Strobe
LDSN = lo;
                                  ! Assert Upper Data Strobe
UDSN = lo:
DBENABLE = hi:
                                 ! Enable Data Bus
next;
                                  ! Execute Pending Assignments
PHI1 = lo;
                                  ! Phase 2
                                  ! Of Clock Cycle 25
PHI2 = hi;
                                  ! Execute Pending Assignments
next;
T = 26;
                                  ! Clock Cycle 26
next;
                                  ! Execute Assignment
                                  ! Phase 1
PHI1 = hi;
                                  ! Of Clock Cycle 26
PHI2 = 10;
while DTACKN eql hi
                                  ! Wait For Memory To Place
                                  ! Data On The Bus
                                  ! Execute Impending Assignments
     next;
     PHI1 = 10;
                                  ! Phase 2
                                  ! Of Clock Cycle 26
     PHI2 = hi;
     next:
                                  ! Execute Assignments
     T = 27:
                                  ! Clock Cycle 27
                                  ! Execute Assignment
```

next;

```
PHI1 = hi:
                               ! Phase 1
    PHI2 = 10;
                               ! Of Clock Cycle 27
    DBUS(15:8> = MEARUS];
                              ! Memory Places Instruction
     DBUS<7:0> = MCABUS + 13;
                               ! On Data Bus And
    INTACKN = lo;
                               ! Asserts DTACKN(Added)
    next;
                               ! Execute Pending Assignments
    T = 26
                               ! Return To Phase 2
                               ! Of Clock Cycle 26
    );
                              ! Execute Impending Assignments
     next:
T = 27:
                                ! Clock Cycle 27
next;
                               ! Execute Assignment
FHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 27
EXDRUF = DRUS;
                               ! Instruction On Data Bus
                               ! Is Placed In External Data
                               ! Bus Buffer
next;
                               ! Execute Pending Assignments
T = 28:
                               ! Clock Cycle 28
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
FHI2 = 10;
                               ! Of Clock Cycle 28
HANADRLOW = EXUBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Handler Routine Low Address
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 28
ASN = hi:
                               ! Neactivate Address Strobe
LUSN = hi;
                               ! Deactivate Lower Data Strobe
UDSN = hi:
                               ! Deactivate Upper Data Strobe
DTACKN = h1;
                               ! Deactivate Data Transfer(Added)
                               ! Acknowledge
VECADR = VECADR + 2;
                               ! Increment Vector Address Register
                               ! To Pick Handler Address Low Word
next:
                               ! Execute Pending Assignments
T = 29:
                                ! Clock Cycle 29
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 24
```

```
RW = hi;
                                   ! Memory Read
ADENABLE = 10;
                                   ! Disable Address Bus Buffer
ABUS = 0xffffff;
                                   ! Address Bus High Impedanced
INSENABLE = 10;
                                   ! Disable Data Bus Buffer
DBUS = 0xffff;
                                   ! Data Bus High Impedanced
IABUS = VECADR;
                                   ! Place VECADR On Internal Address
IDBUS = HANADRLOW;
                                   ! Move Handler High Address To
                                   ! Data Bus
                                   ! Execute Pending Assignments
next;
                                   ! Phase 2 Of
PHI1 = 10;
                                   ! Clock Cycle 29
PHI2 = hi;
ADENABLE = hi:
                                   ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>;
                                   ! Gate Internal Address Bus
                                   ! Into External Address Buffer
FCMODE = SRMODE;
                                   ! Supervisor Mode
FCSPACE = 1;
                                   ! Accessing Data
HANADRHI = IDBUS;
                                   ! Hove Handler High Address
                                   ! To Upper Word Of Register
                                   ! Place Address On External Bus
ARUS = IARUS<23:1>;
next:
                                   ! Execute Impending Assignments
T = 30;
                                    ! Clock Cycle 30
next;
                                   ! Execute Assignment
FHII = hi;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 30
ASN = lo;
                                   ! Assert Address Strobe
                                   ! Assert Lower Data Strobe
LDSN = 10;
UDSN = lo:
                                   ! Assert Upper Data Strobe
                                   ! Enable Data Bus
DBENABLE = hi:
next;
                                   ! Execute Pending Assignments
                                   ! Phase 2
PHI1 = lo;
PHI2 = hi;
                                   ! Of Clock Cycle 30
next;
                                   ! Execute Pending Assignments
T = 31;
                                   ! Clock Cycle 31
next;
                                   ! Execute Assignment
fHII = hi;
                                   ! Phase 1
fHI2 = 10;
                                   ! Of Clock Cycle 31
                                   ! Wait For Memory To Place
while DTACKN eql hi
                                   ! Data On The Bus
                                   ! Execute Impending Assignments
     next;
     fHI1 = lo;
                                   ! Phase 2
     PHI2 = hi;
                                   ! Of Clock Cycle 31
     next;
                                   ! Execute Assignments
```

```
/*********************
    T = 32:
                               ! Clock Cycle 32
    next;
                              ! Execute Assignment
    PHI1 = hi;
                              ! Phase 1
    PH12 = 10;
                              ! Of Clock Cycle 32
    DRUS(15:8) = MCABUS];
                              ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                               ! On Data Bus And
    DITACKN = 10:
                               ! Asserts ITACKN(Added)
    next;
                               ! Execute Pending Assignments
     T = 31
                               ! Return To Phase 2
                               ! Of Clock Cycle 31
    );
    next;
                               ! Execute Impending Assignments
T = 32;
                               ! Clock Cycle 32
next;
                               ! Execute Assignment
FHI1 = 10;
                               ! Phase 2
PHI2 = hi:
                              ! Of Clock Cycle 32
EXDRUF = DRUS;
                               ! Instruction On Nata Bus
                               ! Is Placed In External Data
                               ! Rus Buffer
next;
                              ! Execute Pending Assignments
T = 33;
                               ! Clack Cycle 33
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 33
PHI2 = 10;
HANADRLOW = EXTIBUF;
                               ! The Contents Of The External
                               ! Nata Rus Ruffer Are Placed
                               ! In Handler Routine Low Address
next;
                              ! Execute Pending Assignments
PHI1 = 10;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 33
ASN = hi;
                               ! Deactivate Address Strobe
LDSN = hi;
                               ! Deactivate Lower Data Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
DTACKN = hi;
                              ! Deactivate Data Transfer(Added)
                               ! Acknowledge
next;
                               ! Execute Pending Assignments
T = 34;
                               ! Clock Cycle 34
                               ! Execute Assignment
next;
```

```
! Phase 1 Of
PHI1 = hi;
                                   ! Clock Cycle 34
PHI2 = 10;
RW = hi;
                                   ! Memory Read
ANENABLE = 10;
                                   ! Disable Address Bus Buffer
ABUS = 0xffffff;
                                   ! Address Bus High Impedanced
INENABLE = 10;
                                   ! Disable Data Bus Buffer
DBUS = 0xffff;
                                   ! Data Bus High Impedanced
                                   ! Place HANADE On Internal Address
IABUS = HANADR:
                                   ! Execute Pending Assignments
next;
                                   ! Phase 2 Of
PHI1 = 10;
                                   ! Clock Cycle 34
PHI2 = hi;
ADENABLE = hi;
                                   ! Enable Address Bus Buffer
EXABUF = IABUS<23:1>
                                   ! Gate Internal Address Bus
                                   ! Into External Address Buffer
FCMODE = SRMODE:
                                   ! User Mode
PC = IABUS:
                                   ! Place HANADR In PC
                                . ! Accessing Program
FCSPACE = 2:
ABUS = IABUS<23:1>;
                                   ! Place Address On External Bus
next;
                                   ! Execute Impending Assignments
T = 35;
                                    ! Clock Cycle 35
                                   ! Execute Assignment
next;
                                   ! Phase 1 Of
PHI1 = hi;
PHI2 = 10;
                                   ! Clock Cycle 35
ASN = lo;
                                   ! Assert Address Strobe
LIISN = lo;
                                   ! Assert Lower Data Strobe
UIISN = lo:
                                   ! Assert Upper Data Strobe
DBENABLE = hi;
                                   ! Enable Data Bus
next;
                                   ! Execute Pending Assignments
FHI1 = 10;
                                   ! Phase 2
PHI2 = hi;
                                   ! Of Clock Cycle 35
                                   ! Execute Pending Assignments
next;
! Clock Cycle 36
T = 36;
next;
                                   ! Execute Assignment
                                   ! Phase 1
PHI1 = hi;
PHI2 = 10;
                                   ! Of Clock Cycle 36
                                   ! Wait For Memory To Place
while DTACKN eql hi
                                   ! Data On The Bus
     next;
                                   ! Execute Impending Assignments
     PHI1 = lo;
                                   ! Phase 2
     PHI2 = hi;
                                   ! Of Clock Cycle 36
                                   ! Execute Assignments
     next;
```

```
T = 37;
                                  ! Clock Cycle 37
     next;
                                 ! Execute Assignment
     PHI1 = hi;
                                 ! Phase 1
     FHI2 = 10;
                                 ! Of Clock Cycle 37
     DBUS<15:8> = MEABUS];
                                 ! Memory Places Instruction
     IGRUS <7:0> = MEABUS + 13;
                                 ! On Data Bus And
     DITACKN = 10;
                                 ! Asserts DTACKN(Added)
     next;
                                 ! Execute Pending Assignments
     T = 36
                                  ! Return To Phase 2
                                 ! Of Clock Cycle 36
     );
     next;
                                 ! Execute Impending Assignments
T = 37;
                                  ! Clock Cycle 37
next;
                                 ! Execute Assignment
FHI1 = 10;
                                 ! Phase 2
                                 ! Of Clock Cycle 37
PHI2 = hi;
EXDRUF = DBUS;
                                 ! Instruction On Data Bus
                                 ! Is Placed In External Data
                                 ! Bus Buffer
next;
                                 ! Execute Pending Assignments
T = 38;
                                  ! Clock Cycle 38
next:
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
PHI2 = lo:
                                 ! Of Clock Cycle 38
                                 ! The Contents Of The External
PFR = EXDBUF;
                                 ! Data Bus Buffer Are Placed
                                 ! In Prefetch Register
next;
                                 ! Execute Pending Assignments
PHI1 = 10;
                                 ! Phase 2
PH12 = hi;
                                 ! Of Clock Cycle 38
ASN = hi:
                                 ! Deactivate Address Strobe
LUSN = hi;
                                 ! Deactivate Lower Data Strobe
UDSN = hi;
                                 ! Deactivate Upper Data Strobe
IR = PFR;
                                 ! Contents Of Prefetch Register
                                 ! Are Placed Into Instruction
                                 ! Register
ITACKN = hi;
                                 ! Deactivate Data Transfer(Added)
                                 ! Acknowledge
PC = PC + 2;
                                 ! Increment Program Counter
next;
                                 ! Execute Pending Assignments
```

```
! Clock Cycle 39
     T = 39;
     next;
                                          ! Phase 1 Of
     PHI1 = hi;
                                          ! Clock Cycle 39
     PHI2 = 10;
                                          ! Disable Address Bus Buffer
     ADENABLE = lo;
                                          ! Disable Data Bus Buffer
     DRENABLE = 10;
                                          ! Data Bus High Impedanced
     DBUS = 0xffff;
     ASN = hi;
                                          ! Disable Address,
     LDSN = hi;
                                          ! Lower Data, and
     UDSN = hi;
                                          ! Upper Data Strobes
                                          ! Execute Pending Assignments
     next;
                                          ! Phase 2 Of
     PHI1 = lo;
                                          ! Clock Cycle 39
     PHI2 = hi;
     ABUS = 0xffffff;
                                          ! ABUS High Impedanced
                                          ! Execute Impending Assignments
     next;
     ! Clock Cycle 40
     T = 40;
                                          ! Execute Assignment
     next;
                                          ! Phase 1 Of
     PHI1 = hi;
     PHI2 = 10;
                                          ! Clock Cycle 40
     next;
                                          ! Execute Pending Assignments
                                          ! Phase 2
     PHI1 = lo;
                                          ! Of Clock Cycle 40
     PHI2 = hi;
     next;
                                          ! Reset Clock Cycle Counter
     T = 0
     )
rte :=
   SA7 = SA7 - 2;
   next:
   SR<15:8> = MESA73;
   SR<7:0> = MCSA7 + 13;
   next;
   SA7 = SA7 + 2;
   next;
   PC<31:24> = MESA73;
   PC<23:16> = MESA7 + 13;
   next;
   SA7 = SA7 + 2;
   next:
   PC<15:8> = MESA73;
   PC<7:0> = MESA7 + 13;
   next;
   IR<15:8> = MCPC3;
   IR<7:0> = MEFC + 13;
   next;
```

```
PC = PC + 2;
   T = 20;
   next;
   T = 0
                                         ! MOVE.W D1,D2
move :=
     ! Phase 1 Of
     FHI1 = hi;
                                         ! Clock Cycle 0
     FHI2 = 10;
     DBUS = 0xffff;
                                         ! Place Data Bus In High Impedance
                                         ! Memory Read
     RW = hi;
                                         ! Disable Address Bus Buffer
     ADENABLE = 10;
     DENABLE = 10;
                                         ! Disable Data Bus Buffer
                                         ! Place PC On Internal Address
     IABUS = PC;
                                         | Rus
     IDBUS = DILWORD;
                                         ! Place Low Word From DE13 Onto
                                         ! Internal Data Bus
                                         ! Execute Pending Assignments
     next;
                                         ! Phase 2 Of
     PHI1 = 16:
                                         ! Clock Cycle 0
     PHI2 = hi:
                                         ! Enable Address Bus Buffer
     ADENABLE = hi;
     EXABUF = IABUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
                                         ! User Mode
     FCMODE = SRMODE;
     FCSPACE = 2;
                                         ! Accessing Program
     SRCARRY = lo;
                                         ! Clear Status Register Carry Bit
     SROVER = lo;
                                         ! Clear Status Register Overflow Bit
     SRZERO = lo;
                                         ! Clear Status Register Zero Bit
     SANEG = 10;
                                         ! Clear Status Register Negative Bit
                                         ! Place Data From Internal Data Bus
     D2LWORD = IDBUS;
                                         ! Into Low Word Of D[2]
                                         ! Execute Impending Assignments
     next;
     ABUS = EXABUF;
                                         ! Address Floced On Bus(Added)
                                         ! Execute Pending Assignments
     next;
     ! Clock Cycle 1
     T = 1;
     next;
                                         ! Execute Assignment
     PHI1 = hi;
                                         ! Phase 1 Of
                                         ! Clock Cycle 1
     PHI2 = 10:
     ASN = lo;
                                         ! Assert Address Strobe
     LUSN = 10;
                                         ! Assert Lower Data Strobe
                                         ! Assert Upper Data Strobe
     UDSN = 10;
                                         ! Enable Data Rus
     DBENABLE = hi;
                                         ! Set Status Register Zero Bit
      if I/2LWORD eq1 0
                                         ! If Moved Data Is Zero
        SRZERO = hi;
                                         ! Execute Pending Assignments
     next;
```

```
FHI1 = 10;
                             ! Phase 2
PHI2 = hi:
                             ! Of Clock Cycle 1
if DE23<15>
                             ! Set Status Register Negative
  SRNEG = hi;
                             ! Rit If Moved Data Is Negative
                             ! Execute Pending Assignments
next;
! Clock Cycle 2
next;
                             ! Execute Assignment
PHI1 = hi:
                             ! Phase 1
PHI2 = 10;
                             ! Of Clock Cycle 2
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Data On The Bus
    next;
                             ! Execute Impending Assignments
    PH11 = 10;
                             ! Phase 2
    PHI2 = hi;
                             ! Of Clock Cycle 2
                             ! Execute Assignments
    next;
    T = 3;
                             ! Clock Cycle 3
    next;
                             ! Execute Assignment
    PHI1 = hi;
                             ! Phase 1
    PHI2 = 10:
                             ! Of Clock Cycle 3
    DBUS<15:8> = MCABUS];
                             ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                             ! On Rata Rus And
    DITACKN = lo;
                             ! Asserts DTACKN(Added)
    next;
                             ! Execute Pending Assignments
    1 = 2
                             ! Return To Phase 2
                             ! Of Clock Cycle 2
    );
    next;
                             ! Execute Impending Assignments
T = 3:
                             ! Clock Cycle 3
next;
                             ! Execute Assignment
                             ! Phase 2
FHII = 10;
PHI2 = hi;
                             ! Of Clock Cycle 3
EXDBUF = DBUS;
                             ! Instruction On Data Bus
                             ! Is Placed In External Data
                             ! Bus Buffer
next;
                             ! Execute Pending Assignments
T = 4;
                             ! Clock Cycle 4
next;
                             ! Execute Assignment
```

```
! Phase 1
     PHI1 = hi;
                                         ! Of Clock Cycle 4
     FHI2 = 10;
     PFR = EXPRUF;
                                        ! The Contents Of The External
                                        ! Data Bus Buffer Are Placed
                                         ! In Prefetch Register
                                         ! Execute Pending Assignments
     next;
     PHI1 = 10;
                                         ! Phase 2
                                        ! Of Clock Cycle 4
     PHI2 = hi;
     ASN = hi;
                                         ! Deactivate Address Strobe
     LUSN = hi;
                                         ! Neactivate Lower Data Strobe
     UDSN = hi;
                                         ! Deactivate Upper Data Strobe
     IR = PFR;
                                         ! Contents Of Frefetch Register
                                         ! Are Placed Into Instruction
                                         ! Register
     DTACKN = hi;
                                         ! Deactivate Data Transfer(Added)
                                         ! Acknowledge
     PC = PC + 2;
                                         ! Increment Program Counter
     next;
                                         ! Execute Impending Assignments
     T = 0
                                         ! Reset Clock Cycle Counter
                                         ! JMF (AO)
.imp :=
     PHI1 = hi;
                                         ! Phase 1 Of
                                         ! Clock Cycle 0
     FHI2 = 10;
     DBUS = 0xffff;
                                         ! Place Data Bus In A High Impedance
     RW = hi;
                                         ! Memory Read
     ADENABLE = lo;
                                         ! Disable Address Bus Buffer
                                         ! Disable Data Bus Buffer
     IIBENABLE = 10;
                                         ! Place PC On Internal Address
     IABUS = PC;
                                         Hus
     next;
                                         ! Execute Pending Assignments
                                         ! Phase 2 Of
     f'Hl1 = lo;
                                         ! Clock Cycle 0
     PHI2 = hi;
                                         ! Enable Address Bus Buffer
     ALIENABLE = hi;
     EXABUF = IABUS;
                                         ! Gate Internal Address Bus
                                         ! Into External Address Buffer
     FCMODE = SKMODE:
                                         ! User Mode
     FCSFACE = 2;
                                         ! Accessing Program
                                         ! Execute Pending Assignments
     next;
     ABUS = EXABUF;
                                         ! Address Flaced On Bus(Added)
     next;
                                         ! Execute Pending Assignments
     ! Clock Cycle 1
     T = 1;
                                         ! Execute Assignment
    · next;
```

```
! Phase 1 Of
PHI1 = hi;
                               ! Clock Cycle 1
FHI2 = 10;
ASN = 10;
                               ! Assert Address Strobe
                               ! Assert Lower Data Strobe
LISN = lo;
UIISN = 10:
                               ! Assert Upper Data Strobe
IANUS = ALO];
                               ! Move Jump Address From A[0]
                               ! To Internal Address Buffer
DRENABLE = hi;
                               ! Enable Data Bus
next;
                               ! Execute Fending Assignments
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 1
PH12 = hi;
FC = IABUS;
                               ! Place Jump Address Into Program
                               ! Counter
next;
T = 2;
                               ! Clock Cycle 2
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 2
PHI2 = 10;
                               ! Wait For Memory To Place
while DTACKN eql hi
                               ! Nata On The Bus
    next;
                               ! Execute Impending Assignments
                               ! Phase 2
     PHI1 = lo;
     PHI2 = hi;
                               ! Of Clock Cycle 2
                               ! Execute Assignments
     next;
     ! Clock Cycle 3
                               ! Execute Assignment
     next;
     PHI1 = hi:
                               ! Phase 1
     PHI2 = 10;
                               ! Of Clock Cycle 3
     IDBUS<15:8> = MEABUS3;
                               ! Memory Places Instruction
     DBUS(7:0) = MEABUS + 13;
                               ! On Data Bus And
     ITACKN = 10;
                               ! Asserts DTACKN(Added)
     next;
                               ! Execute Pending Assignments
     T = 2
                               ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
                               ! Execute Impending Assignments
     next;
T = 3:
                               ! Clock Cycle 3
                               ! Execute Assignment
next;
PHI1 = lo;
                               ! Phase 2
```

```
PHI2 = hi;
                                 ! Of Clock Cycle 3
EXUBUF = DBUS;
                                 ! Instruction On Data Bus
                                 ! Is Placed In External Data
                                 ! Rus Buffer
next;
                                 ! Execute Pending Assignments
T = 4:
                                ! Clock Cycle 4
next;
                                 ! Execute Assignment
PHI1 = hi:
                                ! Phase 1
FHI2 = 10;
                                ! Of Clock Cycle 4
next;
PFR = EXDBUF;
                                ! The Contents Of The External
                                ! Data Bus Buffer Are Placed
                                 ! In Prefetch Register
                                ! Execute Pending Assignments
next;
FHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 4
ASN = hi;
                                ! Deactivate Address Strobe
LDSN = hi;
                                ! Deactivate Lower Data Strobe
UIISN = hi;
                                ! Deactivate Upper Data Strobe
DTACKN = hi;
                                 ! Deactivate Data Transfer
                                 ! Acknowledge(Added)
T = 5;
                                 ! Clock Cycle 5
next;
                                 ! Execute Previous Assignment
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 16;
                                 ! Clock Cycle 5
RW = hi;
                                 ! Memory Read
ADENABLE = 10;
                                ! Disable Address Bus Buffer
DBENABLE = 10;
                                 ! Disable Data Bus Buffer
1ABUS = PC:
                                 ! Place PC On Internal Address
                                 ! Bus
next;
                                 ! Execute Pending Assignments
PHI1 = Io;
                                ! Phase 2 Of
FHI2 = hi;
                                ! Clock Cycle 5
AVENABLE = h1;
                                ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                ! User Mode
FCSPACE = 2;
                                ! Accessing Program
EXABUF = TABUS;
                                ! Gate Internal Address Bus
next:
                                ! Into External Address Buffer
ABUS = EXABUF;
                                ! Address Flaced On Bus(Added)
                                ! Execute Pending Assignments
next;
T = 6:
                                ! Clock Cycle 6
next;
                                 ! Execute Assignment
```

```
FHI1 = hi;
                               ! Phase 1 Of
PHI2 = 10;
                               ! Clock Cycle 6
                               ! Assert Address Strobe
ASN = 10;
                               ! Assert Lower Data Strobe
LISN = lo;
UDSN = lo;
                               ! Assert Upper Nota Strobe
DRENABLE = hi:
                               ! Enable Nata Rus
next:
                               ! Execute Pending Assignments
PHI1 = lo:
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 6
next;
                               ! Execute Pending Assignments
T = 7;
                               ! Clock Cycle 7
next;
                               ! Execute Assignment
                               ! Phase 1
PHI1 = hi;
                               ! Of Clock Cycle 7
PHI2 = 10:
                               ! Wait For Memory To Place
while DTACKN eql hi
                               ! Data On The Bus
     next;
                               ! Execute Impending Assignments
                               ! Phase 2
     PHI1 = lo;
     PHI2 = hi;
                               ! Of Clock Cycle 7
     next;
                               ! Execute Assignments
     T = 8:
                               ! Clock Cycle 8
     next;
                               ! Execute Assignment
                               ! Phase 1
     PHI1 = hi;
                               ! Of Clock Cycle 8
     PHI2 = 10;
     DBUS<15:8> = MEABUS];
                               ! Memory Places Instruction
                               ! On Data Rus And
     I(RUS<7:0) = MEARUS + 13;
     ITACKN = lo;
                               ! Asserts DTACKN(Added)
     next;
                               ! Execute Pending Assignments
     ! Return To Phase 2
     T = 7
                              ! Of Clock Cycle 7
     );
                              ! Execute Impending Assignments
     next;
T = 8;
                               ! Clock Cycle B
next;
                               ! Execute Assignment
FHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 8
EXDRUF = DRUS;
                               ! Instruction On Bata Rus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
```

```
T = 9;
                                          ! Clock Cycle 9
     next;
                                          ! Execute Assignment
     PHI1 = hi;
                                          ! Phase 1
     PHI2 = 10;
                                          ! Of Clock Cycle 9
     PFR = EXDBUF;
                                          ! The Contents Of The External
                                          ! Data Bus Buffer Are Placed
                                          ! In Frefetch Register
     next;
                                          ! Execute Pending Assignments
     PHI1 = lo;
                                          ! Phase 2
     PHI2 = hi;
                                          ! Of Clock Cycle 9
     ASN = hi;
                                          ! Deactivate Address Strobe
     LDSN = hi;
                                          ! Deactivate Lower Data Strobe
     UDSN = hi;
                                          ! Deactivate Upper Data Strobe
     PC = PC + 2;
                                          ! Increment Program Counter
     IR = PFR;
                                          ! Place Contents Of Prefetch
                                          ! Register Into Instruction
                                          ! Register
     DTACKN = hi:
                                          ! Deactivate Data Transfer
                                          ! Acknowledge(Added)
     next;
                                          ! Execute Pending Assignments
     T = 0
                                          ! Reset Clock Cycle Counter
decode_execute_prefetch :=
                       case IR
                           032001: move
                                          ! MOVE.W D1,D2
                           047320: jmp
                                          ! JMF (AO) If IR = Octal Value
                           0x4e73: rte
                                          ! RTE (Return From Exception)
                           0x4afc: illegal! Illegal Instruction
                       esac
                       )
main :=
    power on initialize;
    fetch_imitial_instruction;
    while READY eql hi
          decode_execute_prefetch
    )
```

```
/×
                                               11/
/#
    MOTORULA MC68000 MODEL OF THE ILLEGAL ADDRESS EXCEPTION
                                               */
/x
                                               1/
/*
                                               */
/*
              Structure Declarations
                                               */
/*
                                               */
state
1/
/*
/*
                                               */
           M68000 Programming Registers
                                               */
/*
DE0:73<31:0>,
                  ! 8 Data Registers
AE0:63<31:0>,
                  ! 7 Address Registers
UA7<31:0>,
                  ! User Stack Pointer
SA7<31:0>,
                  ! System Stack Pointer
PC<31:0>,
                  ! Program Counter'
SR<15:0>,
                  ! Status Register
/*
                                               */
/*
                                               */
           Temporary Internal Registers
                                               4/
/*
PFR<15:0>,
                  ! Frefetch Register
IR<15:0>,
                  ! Instruction Register
FC<2:0>,
                  ! Function Code Register
                  ! External Data Bus Buffer Register
EXDBUF<15:0>,
EXABUF <23:1>,
                  ! External Address Bus Buffer Register(changed)
ALUBUF1<31:0>,
                  ! ALU Buffer 1
                  ! ALU Buffer 2
ALUBUF2<3110>,
DITEMPK15:0>,
                  ! Temporary Data Storage
DISREG<31:0>.
                  ! Temporary Displacement Storage
SRTEMP<15:0>,
                  ! Temporary Status Register Storage
                  ! (Exception Processing)
IRTEMP<15:0>.
                  ! Temporary Instruction Register Storage
                  ! (Exception Processing)
TEMPADR<31:0>.
                  ! Temporary Cycle Address Storage
                  ! (Exception Processing)
ACTYPE<15:0>,
                  ! Temporary Access Type Storage
                  ! (Exception Processing)
VECADR<23:0>,
                  ! Temporary Vector Address Storage
                  ! (Exception Processing)
```

```
HANADR<31:0>,
                        ! Temporary Address Storage For
                        ! Exception Handler Routine
T<7:0>.
                        ! Clock Cycle Counter
RESET,
                     ! Reset Flip-Flop
HALT,
                     ! Halt Flip-Flop
Ŕ₩,
                     ! Read/Write Flip-Flop
ADENABLE,
                     ! Address Bus Buffer Enable
DIRENABLE,
                     ! Nata Bus Buffer Enable
ASN.
                     ! Address Strobe Flip-Flop
LIISN.
                     ! Lower Nata Strobe Flip-Flop
UDSN.
                     ! Upper Data Strobe Flip-Flop
DITACKN,
                     ! Natu Transfer Acknowledge Flip-Flop
COUT,
                     ! Carry Flip-Flop
EXCEPT.
                     ! Exception Processing Flip-Flop
READY,
                     ! Ready Flip-Flop
/*
/*
       Model transformation modifications:
                                                             */
/*
                                                             1/
/*
           1) CDL decoder structure nonexistent in ISP' and un-
                                                             */
/x
       necessary for model. Eliminated.
                                                             */
/*
                                                             */
           Multi-phase clock structure nonexistent in ISP'.
/*
       Operations on registers will provide its equivalent.
                                                             */
/*
           3) Switch structure nonexistent in ISP'. Operation on a
                                                             */
/*
                                                             */
       register will provide its equivalent.
/*
           4) The declared bus structures are modeled with registers */
/*
       without loss of model accurracy. This done to maintain model
                                                             */
/*
       equivalency and simplicity.
                                                             */
/x
           5) The memory word length was reduced from 16 to 8 bit
                                                             1/
/x
       words to coincide with the ECR's 32-Kbyte memory, to agree with*/
1*
       their PC incrementation, and to enable the use of existing
                                                             */
/x
       MC68000 assembler and linker/loader models. The memory was
                                                             */
/*
       also reduced from 8 Mwords to 32 Nbytes.
                                                             */
/*
                                                             */
TABUS<31:0>,
                        ! Internal Address Bus
                        ! Internal Data Bus
IDRUS<31:0>,
twait<7:0>,
                        ! Wait Cycle Counter
SWITCH.
                     ! Power Switch
                     ! Phase 1 Of Two-fhase Clock
PHI1,
PHI2:
                     ! Phase 2 Of Two-Phase Clock
port
/*
/*
             External Address and Data Bus
                                                             */
/*
                                                             1/
DBUS<15:0>,
                       ! External Data Bus
```

```
ABUS<23:1>;
                   ! External Address Rus(changed)
format
/*
                                                        */
/*
                Register Subfields
                                                        */
                                                        */
/*
PCADDR
         = PC(23:0),
                      ! Fragram Counter Address Field
SRTRACE
         = SR<15>,
                      ! Trace Bit
SKHODE
         ≈ SR<13>,
                      ! Mode Selection Bit
SRCARRY
                      ! Carry Bit
         = Sk<0>,
SROVER
         = SR<1>.
                      ! Overflow Bit
SRZERO
         = SR<2>,
                      ! Zero Rit
SRNEG
         = SR<3>.
                      ! Negative Bit
SREX
         = SR<4>,
                      ! Extend Bit
SRMASK
                      ! Interrupt Mask
         = SR<10:8>
FCSHACE
         = FC<1:0>,
                      ! Memory Access Address Space
FCMODE
         = FC<2>.
                      ! User/Supervisor Mode Bit
F'CLUW
         = PC<15:0>,
                      ! PC Low Word
PCHI
         = PC(31:16),
                      ! PC High Word
DOLWORD
         = DC03<15:0>,
                      ! DEOJ Low Word
DILWORD
                      ! D[1] Low Word
         = D[1](15:0),
D2LWORD
                      ! DE23 Low Word
         = BC23<15:0>,
D3LWORD
         = D(3)(15:0).
                      ! DE3] Low Word
I/4LWORD
         = D[4]<15:0>,
                      ! D[43 Low Word
DSLWORD
         = DESI<15:0>,
                      ! D[5] Low Word
                     ! DE63 Low Word
II6LWORD
         = D[6](15:0),
         = D[7]<15:0>,
                      ! DE73 Low Word
D7LWORD
DISREGHWORD = DISREG<31:16>,! DISREG High Word
DISREGLWORD = DISREG<15:0>, ! DISREG Low Word
HANADRLOW
         = HANADR<15:0>, ! HANADR Low Word
HANADRHI
         = HANADR<31:16>,! HANADR High Word
TEMPADRLOW = TEMPADR<15:0>,! TEMPADR Low Word
TEMPAURHI
         = TEMPADR<31:16>:! TEMPADR High Word
Memory
/*
/*
                16K 16-kit Word Internal Memory
                                                        */
/*
                                                        1/
HE0:327673<7:0>;
mac 1.0
*/
/*
/*
               Logic Level Macros
                                                        */
```

```
= 0 1.
l٥
hi
     = 1 %,
     = 0 1,
off
on
     = 1 &,
clear = 0 %;
/*
                                                           */
                                                           */
/* Fower On and Initialization. This process was not modeled but is
                                                           */
  added to initialize signals and registers.
                                                           */
/*
power_on_initialize :=
      SWITCH = on;
                                  ! Turn Power On
                                  ! Execute Assignment
      next;
      READY = 10;
                                  ! System Not Ready
                                  ! Assert Reset For
      RESET = lo;
                                  ! 100 Miliseconds(Active Low)
      delay(100);
      RESET = hi:
                                  ! Deactivate Reset
                                  ! Execute Pending Assignments
      next;
      ASN = hi;
                                  ! Initialize Address Strobe
                                  ! Initialize Lower Data Strobe
      LDSN = hi:
                                  ! Initialize Upper Data Strobe
      UDSN = hi;
                                  ! Initialize Nata Transfer Acknowledge
      IITACKN = hi;
                                  ! Initialize Read/Write(Read On High)
      RW = hi:
                                 ! Place Data Bus In High Impedance State
      IIBUS = Oxffff;
                                 ! Place Memory Locations Following The
      ME0x100aJ = 0xff;
      M[0x100b] = 0xff;
                                  ! JMP Instruction In A High State
                                  ! Initialize Halt Flip-Flop(Active
      HALT = hi;
                                  ! Low)
      T = 0;
                                  ! Initialize Clock Cycle Counter
      READY = hi;
                                  ! System Ready
       /*
                                                           */
      /*
            Routine Initialization Per Hamby and Guillory
                                                           */
       /*
                                                           */
       ! Place Hex 5555555 Into D[1]
      D[1] = 0x55555555;
      A[0] = 0:1000;
                                  ! Place Hex 1000 Into A[0]
                                  ! Flace Illegal Address
      A[1] = 0x2001;
      PC = 0 \times 1000;
                                  ! Place Hex 1000 Into Program Counter
                                 ! Place Exception
      10x0 = [0x0]M
      MEO \times d1 = O \times O:
                                 ! Vector Beginning In
                                 ! Location
      M[0xe] = 0x20;
      M[0xf] = 0x40;
                                 ! C Hex
      SA7 = 0x0786;
                                  ! Initialize System Stack Pointer
                                  ! At Program Headed Down (Added)
```

next

! Execute Assignments

```
/*
/# Initial Fetch Cycle. This cycle was not modeled but is necessary
                                                              */
/* to retrieve modeled instructions for simulation and analysis. It
/* was fashsioned from the Read Cycle described by Hamby and Guillory */
   on page VI-15 of their thesis.
                                                              */
/*
                                                              */
/*
fetch_initial_instruction :=
     ! Phase 1 Of
     PHI1 = hi;
                                       ! Clock Cycle 0
     PHI2 = 10;
                                       ! Memory Read
     RW = hi;
                                       ! Disable Address Bus Buffer
     ADENABLE = lo;
                                       ! Disable Data Bus Buffer
     DBENABLE = 10;
                                       ! Place PC On Internal Address
     IABUS = PC;
                                       ! Rus
                                       ! Execute Pending Assignments
     next;
                                       ! Phase 2 Of
     PHI1 = 10;
                                       ! Clock Cycle 0
     PHI2 = hi:
                                       ! Enable Address Bus Buffer
     ADENABLE = hi;
                                       ! Gate Internal Address Bus
     EXARUF = IABUS:
                                       ! Into External Address Buffer
     FCMODE = SRMODE;
                                       ! User Mode
                                       ! Accessing Program
     FCSPACE = 2;
                                       ! Execute Impending Assignments
     next;
                                       ! Address Placed On Bus(Added)
     ABUS = EXABUF;
                                       ! Execute Pending Assignments
     next;
     ! Clock Cycle 1
     T = 1:
                                       ! Execute Assignment
     next;
     PHI1 = hi;
                                       ! Phase 1 Of
                                       ! Clock Cycle 1
     PHI2 = 10;
     ASN = lo:
                                       ! Assert Address Strobe
     LDSN = lo;
                                       ! Assert Lower Data Strobe
                                       ! Assert Upper Data Strobe
     UDSN = lo;
                                       ! Enable Data Bus
     DBENABLE = hi:
     next;
                                       ! Execute Pending Assignments
     PHI1 = lo;
                                       ! Phase 2
     PHI2 = hi;
                                       ! Of Clock Cycle 1
                                       ! Execute Pending Assignments
     next;
```

)

```
T = 2;
                               ! Clock Cycle 2
                               ! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi:
                               ! Of Clock Cycle 2
PHI2 = 1o:
while DTACKN eql hi
                               ! Wait For Memory To Place
                               ! Data On The Bus
                               ! Execute Impending Assignments
     next;
     PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 2
     PHI2 = hi;
     next;
                               ! Execute Assignments
     ! Clock Cycle 3
                               ! Execute Assignment
     next;
     PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 3
     PH12 = 10;
     DBUS<15:8> = MCABUS3;
                               ! Memory Flaces Instruction
     DBUS<7:0> = MCABUS + 13;
                               ! On Data Rus And
                               ! Asserts DTACKN(Added)
     DTACKN = lo;
                               ! Execute Pending Assignments
     next;
     ! Return To Phase 2
                               ! Of Clock Cycle 2
     );
                               ! Execute Impending Assignments
     next;
! Clock Cycle 3
T = 3;
next:
                               ! Execute Assignment
                               ! Phase 2
FHI1 = lo;
PHI2 = hi;
                               ! Of Clock Cycle 3
EXDRUF = DRUS;
                               ! Instruction On Data Rus
                               ! Is Placed In External Data
                               ! Bus Buffer
                               ! Execute Pending Assignments
next;
T = 4;
                               ! Clock Cycle 4
next;
                               ! Execute Assignment
FHI1 = hi;
                               ! Phase 1
PHI2 = lo;
                               ! Of Clock Cycle 4
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
                               ! Execute Pending Assignments
next;
fHI1 = 10;
                               ! Phase 2
```

```
! Of Clock Cycle 4
     PHI2 = hi;
                                          ! Deactivate Address Strobe
     ASN = hi;
                                          ! Neactivate Lower Data Strobe
     LIISN = hi:
                                           ! Deactivate Upper Data Strobe
     UDSN = hi:
                                           ! Contents Of Prefetch Register
     IR = PFR;
                                           ! Are Placed Into Instruction
                                           ! Register
                                           ! Deactivate Data Transfer(Added)
     DTACKN = hi;
                                           ! Acknowledge
                                           ! Increment Program Counter
     PC = PC + 4;
                                           ! Execute Pending Assignments
     next;
                                           I Reset Clock Cycle Counter
     T = 0
                                           ! AND .W #SDFFF,SR
andi :=
                                           ! Effect Of Instruction
     SEMODE = lo;
                                           ! Prefetch Next Instruction
     IR(15:8> = MEFC3;
     IR<7:0> = MEPC + 13;
                                           ! Is To Set Status Register
     next;
                                              ! Increment Program Counter
        PC = PC + 2;
                                           ! Supervisor Bit To User
     T = 5;
                                           | Mode
     next;
                                           ! Requires & Clock Cycles
     T = 0
     )
                                           ! ADDQ.L $4,A7
uddq :=
                                           ! Effect Of Instruction
     1f SRMODE
                                           ! Is To Increment Either
        SA7 = SA7 + 4
                                           ! System Or User Stack
     else
                                           ! Register By 4 Depending On
        UA7 = UA7 + 4;
                                           ! Prefetch Next Instruction
     IR<15:8> = MCPC3;
     IR<7:0> = MEPC + 13;
                                           ! Is To Set Status Register
     next;
                                           ! Increment Program Counter
     PC = PC + 2;
                                           ! Requires 8 Clock Cycles
     T = 7;
     next:
     T = 0
                                            ! Illegal Address (OAFD)
illegal :=
        ! Phase 1 Of
      PHI1 = hi:
                                            ! Clock Cycle 0
      FHI2 = 10;
                                            ! Memory Read
      RW = hi;
                                            ! Execute Pending Assignments
      next;
                                            ! Phase 2 Of
      PHI1 = 10;
                                            ! Clack Cycle 0
      PHI2 = hi;
```

```
ADENABLE = 10;
                               ! Disable Address Bus Buffer
DRENABLE = 10:
                               ! Disable Data Bus Buffer
DBUS = 0xffff;
                               ! Data Bus High Impedanced
next;
                               ! Execute Fending Assignments
T = 1;
                               ! Clock Cycle 1
                               ! Execute Assignment
next;
FHI1 = ni;
                               ! Phase 1 Of
PH12 = 16;
                               ! Clock Cycle 1
                               ! Flace Status Register On
IDBUS = SR:
                               ! Internal Data Bus
                               ! Execute Pending Assignments
next;
PHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 1
SRIEMP = IDRUS:
                               ! Place Status Register
                               ! On Bus In Temporary Register
                               ! Execute Pending Assignments
next;
T = 2;
                               ! Clock Cycle 2
next;
                               ! Execute Assignment
PHI1 = hi:
                               ! Phase 1
                               ! Of Clock Cycle 2
PHI2 = 10;
SAMODE = hi;
                               ! Set Supervisor State
SRTRACE = 10;
                               ! Turn Off Trace
next;
                               ! Execute Pending Assignments
FHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 4
                               ! Execute Assignment
next:
T = 3;
                               ! Clock Cycle 3
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 3
SA7 = SA7 - 21
                               ! Necrement System Stack Pointer
                               ! To Point To Location That Will
                               ! Receive PC's Low Word
next;
                               ! Execute Pending Assignments
FHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 3
PHI2 = hi;
next;
                               ! Execute Pending Assignments
T = 4;
                               ! Clock Cycle 4
next;
```

```
! Phase 1 Of
FHI1 = hi;
                              ! Clock Cycle 4
PHI2 = 10;
VECAIR = 3;
                              ! Place Vector Number In Register
next;
                              ! Execute Pending Assignments
                              ! Phase 2 Of
PHI1 = 10:
                              ! Clock Cycle 4
PHI2 = hi;
next:
                              ! Execute Impending Assignments
T = 5:
                               ! Clock Cycle 5
next;
                               ! Execute Assignment
                               ! Phase 1 Of
PHI1 = hi:
PHI2 = 10;
                              ! Clock Cycle 5
VECAUR = VECAUR *: arith 2;
                              ! Multiply Vector Number
                              ! By 4 For Vector Address
next;
                              ! Execute fending Assignments
PHI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 5
PHI2 = hi;
next;
                              ! Execute Pending Assignments
T = 6;
                              ! Clock Cycle 6
                              ! Execute Assignment
next;
PHI1 = hi:
                              ! Phase 1
                              ! Of Clock Cycle 6
FH12 = 10;
TEMPADE = EXABUF:
                              ! Save Current Address
next;
                              ! In Temporary Register
PHI1 = 10;
                              ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 6
next;
T = 7:
                               ! Clock Cycle 7
next;
                               ! Execute Assignment
FHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 7
                               ! Execute Pending Assignments
next;
FHI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 7
PHI2 = hi:
next;
T = 8:
                               ! Clock Cycle B
                               ! Execute Assignment
next;
```

```
PHI1 = hi;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 8
RW = hi;
                                   ! Mesory Read
ADENABLE = lo;
                                   ! Disable Address Bus Ruffer
DBENABLE = 10;
                                   ! Disable Data Bus Buffer
LIBUS = Oxffff;
                                    ! Data Rus High Impedanced
IABUS = SA7;
                                    ! Place SA7 On Internal Address
                                   ! Execute Pending Assignments
next;
FHI1 = 10;
                                 · ! Phase 2 Of
                                   ! Clock Cycle 8
PH12 = hi;
                                   ! Enable Address Bus Buffer
ABENABLE = hi;
EXABUF = IABUS;
                                   ! Gate Internal Address Bus
                                   ! Into External Address Buffer
FCMODE = SRMODE;
                                   ! Supervisor Mode
FCSPACE = 1;
                                   ! Accessing Data
IDRUS = PCLOW:
                                   ! Place Low Word from PC Onto
                                   ! Internal Data Bus
                                   ! Execute Impending Assignments
next;
ARUS = EXARUF;
                                   ! Address Flaced On Bus(Added)
next;
                                   ! Execute Pending Assignments
T = 9;
                                   ! Clock Cycle 9
next;
                                   ! Execute Assignment
                                   ! Phase 1 Of
FHI1 = hi;
                                   ! Clock Cycle 9
FHI2 = 10;
ASN = lo:
                                   ! Assert Address Strobe
RW = lo;
                                   ! Activate Write
                                   ! Place Internal Data Bus
EXDBUF = IDBUS;
                                   ! Contents Into External Data Buffer
                                   ! Execute Pending Assignments
next;
                                   ! Phase 2
PHI1 = lo;
PHI2 = hi;
                                   ! Of Clock Cycle 9
DBUS = EXDBUF;
                                   ! Contents Of External Data Buffer
                                   ! Placed On Data Bus
                                   ! Enable Data Bus
DBENABLE = hi;
next;
                                   ! Execute Pending Assignments
7 = 10:
                                    ! Clock Cycle 10
                                   ! Execute Assignment
next;
                                   ! Phase 1
PHI1 = hi;
                                   ! Of Clock Cycle 10
PHI2 = 10;
UIISN = lo;
                                    ! Activate Upper Data Strobe
                                    ! Activate Lower Nata Strobe
LDSN = lo;
twait = 0:
next;
while DTACKN eql hi
                                   ! Wait For Memory To Place
```

```
! Data On The Bus
     twait = twait + 1;
    next:
                               ! Execute Impending Assignments
    PHI1 = lo:
                               ! Phase 2
    FHI2 = hi;
                              ! Of Clock Cycle 10
     next:
                               ! Execute Assignments
     T = 11;
                               ! Clock Cycle 11
     next;
                               ! Execute Assignment
     PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 11
     PHI2 = 10;
     if twait eql 2
                              ! PC Low Word
     MEARUS] = DRUS(15:8);
     MEARUS + 1] = DRUS(7:0);
                               ! Stored
     DITACKN = 10
                              ! Asserts DTACKN(Added)
    );
                              ! Execute Fending Assignments
    next;
     T = 10
                               ! Return To Phase 2
                               ! Of Clock Cycle 10
     );
                              ! Execute Impending Assignments
    next;
T = 11;
                               ! Clock Cycle 11
                               ! Execute Assignment
next;
                               ! Phase 2
PHI1 = 10;
                               ! Of Clock Cycle 11
PHI2 = hi;
SA7 = SA7 - 4;
                               ! Set System Stack Pointer
                              ! To Point To Status Register
                               ! Storage Location
next;
                               ! Execute Pending Assignments
T = 12;
                               ! Clock Cycle 12
                               ! Execute Assignment
next;
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 12
next;
                               ! Execute Pending Assignments
                               ! Phase 2
PHI1 = 10;
                               ! Of Clock Cycle 12
PHI2 = hi;
                               ! Neactivate Address Strobe
ASN = hi;
LUSN = hi;
                               ! Deactivate Lower Data Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
                               ! Deactivate Data Transfer(Added)
DTACKN = hi;
```

```
! Acknowledge
next;
                                 ! Execute Pending Assignments
T = 13;
                                 ! Clock Cycle 13
next;
                                 ! Phase 1 Of
PHI1 = hi;
PHI2 = 10:
                                 ! Clock Cycle 13
KW = hi:
                                 ! Memory Read
ADENABLE = 10;
                                 ! Disable Address Bus Buffer
DEBENABLE = 10:
                                 ! Disable Data Rus Ruffer
IABUS = SA7;
                                 ! Place SA7 On Internal Address
next;
                                 ! Execute Pending Assignments
F'HI1 = lo;
                                 ! Phase 2 Of
PHI2 = hi;
                                 ! Clock Cycle 13
                                 ! Enable Address Bus Buffer
ADENABLE = hi;
                                ! Data Bus High Impedanced
DBUS = 0xffff;
                                ! Gate Internal Address Rus
EXABUF = IABUS;
                                ! Into External Address Buffer
FCMODE = SRMODE:
                                ! Supervisor Mode
FCSFACE = 1;
                                ! Accessing Data
                                ! Place Holder Of Status Register
IDBUS = SRTEMP;
                                ! Onto Internal Data Bus
                                 ! Execute Impending Assignments
next;
ABUS = EXABUF;
                                 ! Address Placed On Bus(Added)
next;
                                 ! Execute Pending Assignments
T = 14:
                                 ! Clock Cycle 14
                                 ! Execute Assignment
next;
                                 ! Phase 1 Of
PHI1 = hi;
PHI2 = 10:
                                 ! Clock Cycle 14
ASN = 10;
                                 ! Assert Address Strobe
RW = lo;
                                 ! Activate Write
EXDBUF = IDBUS;
                                 ! Internal Data Bus Moved
                                 ! To External Data Buffer
                                 ! Execute Pending Assignments
next;
PHI1 = 16;
                                ! Phase 2
PHI2 = hi:
                                ! Of Clock Cycle 14
DBUS = EXDBUF;
                                ! Contents Of External Data
                                ! Buffer Placed On Data Bus
DBENABLE = hi;
                                ! Enable Data Bus
                                 ! Execute Fending Assignments
next;
T = 15:
                                 ! Clock Cycle 15
next;
                                 ! Execute Assignment
```

```
PHI1 = bi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 15
                               ! Activate Upper Data Strobe
UDSN = lo;
                                ! Activate Lower Data Strobe
LDSN = 10;
twait = 0;
next:
while DTACKN eql hi
                               ! Wait For Memory To Place
                               ! Data On The Bus
    twait = twait + 1;
    next;
                               ! Execute Impending Assignments
     PHI1 = 10;
                               ! Phase 2
    PHI2 = hi;
                               ! Of Clock Cycle 15
                               ! Execute Assignments
    next;
     T = 16:
                               ! Clock Cycle 16
    next;
                               ! Execute Assignment
    PHI1 = hi;
                               ! Phase 1
    FH12 = 10;
                               ! Of Clock Cycle 16
     if twait eq1 2
     MCABUS3 = DBUS<15:8>;
                               ! PC Low Word
    MCARUS + 13 = DBUS<7:0>;
                               ! Stored
    IITACKN = 10
                              ! Asserts DTACKN(Added)
    );
    next:
                              ! Execute Fending Assignments
     T = 15
                               ! Return To Phase 2
                               ! Of Clock Cycle 15
     );
                              ! Execute Impending Assignments
    next;
T = 16;
                               ! Clock Cycle 16
                               ! Execute Assignment
next;
                               ! Phase 2
PHI1 = lo;
PHI2 = hi:
                               ! Of Clock Cycle 16
SA7 = SA7 + 2;
                               ! Set System Stack Pointer
                               ! To Point To High PC
                               ! Storage Location
next:
                               ! Execute Pending Assignments
T = 17;
                                ! Clock Cycle 17
                               ! Execute Assignment
next;
PHI1 = hi;
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 17
                               ! Execute Pending Assignments
next;
```

```
FHI1 = 10;
                                   ! Phase 2
                                   ! Of Clock Cycle 17
PHI2 = hi;
ASN = hi;
                                  ! Deactivate Address Strobe
LDSN = hi;
                                   ! Deactivate Lower Data Strobe
UDSN = hi:
                                   ! Deactivate Upper Data Strobe
DTACKN = hi;
                                  ! Deactivate Data Transfer(Added)
                                   ! Acknowledge
next:
                                   ! Execute Pending Assignments
T = 18;
                                   ! Clock Cycle 18
next;
                                   ! Execute Pending Assignments
PHI1 = hi;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 18
RW = hi;
                                   ! Memory Read
                                   ! Disable Address Bus Buffer
ADENABLE = 10;
DIBENABLE = 10;
                                   ! Disable Data Bus Buffer
                                   ! Place SA7 On Internal Address
IABUS = SA7;
                                   ! Bus
next;
                                   ! Execute Pending Assignments
PHI1 = 10;
                                   ! Phase 2 Of
                                   ! Clock Cycle 18
PHI2 = hi;
ADENABLE = hi:
                                   ! Enable Address Bus Buffer
                                  ! Nata Rus High Impedanced
INUS = 0xffff;
                                   ! Gate Internal Address Bus
EXABUF = IABUS;
                                   ! Into External Address Ruffer
FCMODE = SRMODE:
                                   ! Supervisor Mode
FCSPACE = 1:
                                  ! Accessing Data
IDRUS = PCHI;
                                  ! Place High Word Of PC
                                  ! Onto Internal Data Rus
next;
                                   ! Execute Impending Assignments
ABUS = EXABUF;
                                   ! Address Placed On Bus(Added)
next;
                                   ! Execute Pending Assignments
T = 19;
                                   ! Clock Cycle 19
next;
                                   ! Execute Assignment
                                   ! Phase 1 Of
FHI1 = hi;
FHI2 = 10;
                                   ! Clock Cycle 19
                                   ! Assert Address Strobe
ASN = 10;
RW = 10;
                                   ! Activate Write
EXUBUF = IDBUS;
                                   ! Internal Data Bus Moved
                                   ! To External Data Buffer
                                   ! Execute Pending Assignments
next;
FHII = lo;
                                   ! Phase 2
PHI2 = hi;
                                   ! Of Clock Cycle 19
DRUS = EXDRUF;
                                   ! Contents Of External Data
                                   ! Buffer Placed On Data Bus
```

```
DRENABLE = hi;
                               ! Enable Data Bus
next;
                               ! Execute Pending Assignments
T = 20;
                                ! Clock Cycle 20
next;
                               ! Execute Assignment
PHI1 = hi:
                               ! Phase 1
FHI2 = 10:
                               ! Of Clock Cycle 20
UDSN = 10;
                               ! Activate Upper Data Strobe
LISN = lo;
                                ! Activate Lower Data Strobe
twait = 0;
next;
                               ! Wait For Memory To Flace
while DTACKN eql hi
                               ! Data On The Bus
     twait = twait + 1;
                               ! Execute Impending Assignments
    next;
    PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 20
    FHI2 = hi;
    next;
                               ! Execute Assignments
    \x*******************
    T = 21;
                               ! Clock Cycle 21
    next;
                               ! Execute Assignment
                              ! Phase 1
    PHI1 = hi:
                               ! Of Clock Cycle 21
    FH12 = 10;
    if twait eql 2
    M[ABUS] = 1/BUS<15:8>;
                              ! PC Low Word
    MCABUS + 1] = DBUS<7:0>;
                               ! Stored
    DITACKN = 10
                              ! Asserts DTACKN(Added)
    );
    next;
                               ! Execute Pending Assignments
     T = 20
                                ! Return To Phase 2
                               ! Of Clock Cycle 20
    );
                               ! Execute Impending Assignments
    next;
T = 21:
                               ! Clock Cycle 21
next;
                               ! Execute Assignment
PHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 21
SA7 = SA7 - 4;
                               ! Set System Stack Pointer
                               ! To Point To Saved IR
next;
                               ! Execute Pending Assignments
```

/***************************

```
T = 22:
                                    ! Clock Cycle 22
next;
                                   ! Execute Assignment
PHI1 = hi:
                                   ! Phase 1
PHI2 = 10;
                                   ! Of Clock Cycle 22
next;
                                   ! Execute Pending Assignments
PHI1 = 10;
                                   ! Phase 2
                                   ! Of Clock Cycle 22
PHI2 = hi:
ASN = hi:
                                   ! Deactivate Address Strobe
LDSN = hi:
                                   ! Deactivate Lower Data Strope
                                   ! Deactivate Upper Data Strobe
UUSN = bi:
                                   ! Deactivate Data Transfer(Added)
DITACKN = hi;
                                   ! Acknowledge
                                   ! Execute Pending Assignments
next;
T = 23:
                                   ! Clock Cycle 23
                                   ! Execute Pending Assignments
next;
PHI1 = hi;
                                   ! Phase 1 Of
                                   ! Clock Cycle 23
PHI2 = 10;
                                   ! Hemory Read
RW = hi;
                                   ! Disable Address Bus Buffer
ADENABLE = lo;
                                   ! Disable Data Bus Buffer
IBENABLE = 10;
IABUS = SA7;
                                   ! Place SA7 On Internal Address
                                   Hus
next;
                                   ! Execute Pending Assignments
PHI1 = lo;
                                   ! Phase 2 Of
                                   ! Clock Cycle 23
PHI2 = hi;
                                   ! Enable Address Bus Buffer
ADENABLE = hi:
DBUS = 0xffff;
                                   ! Data Bus High Impedanced
EXABUF = IABUS;
                                   ! Gate Internal Address Bus
                                   ! Into External Address Buffer
                                   ! Supervisor Mode
FCMODE = SRMODE;
FCSPACE = 1:
                                   ! Accessing Data
IDBUS = IRTEMP;
                                   ! Place IR Causing Address Error
                                   ! Onto Internal Data Bus
next;
                                   ! Execute Impending Assignments
ABUS = EXABUF;
                                   ! Address Flaced On Rus(Added)
                                   ! Execute Pending Assignments
next:
T = 24:
                                    ! Clock Cycle 24
next;
                                   ! Execute Assignment
PHI1 = hi;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Lycle 24
ASN = lo:
                                   ! Assert Address Strobe
RW = 10;
                                   ! Activate Write
                                   ! Internal Data Bus Moved
EXDBUF = IDBUS;
```

! To External Data Buffer

```
! Execute Pending Assignments
next;
PHI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 24
PHI2 = hi;
DIBUS = EXDBUF;
                              ! Contents Of External Data
                              ! Buffer Placed On Data Bus
DBENABLE = hi:
                              ! Enable Data Bus
next;
                              ! Execute Pending Assignments
T = 25;
                              ! Clock Cycle 25
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1
PHI2 = 10;
                              ! Of Clock Cycle 25
UDSN = 10;
                              ! Activate Upper Data Strobe
                              ! Activate Lower Nata Strobe
LDSN = 10;
twait = 0:
next;
                             ! Wait For Memory To Place
while DTACKN eql hi
                              ! Nata On The Bus
    twait = twait + 1;
                             ! Execute Impending Assignments
    next;
    PHI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 25
    PHI2 = hi;
    next;
                              ! Execute Assignments
     T = 26;
                              ! Clock Cycle 26
                             ! Execute Assignment
    next;
     PHI1 = hi;
                              ! Phase 1
                              ! Of Clock Cycle 26
    PHI2 = 10;
     if twait eql 2
    MEABUS3 = DBUS<15:8>;
                             ! PC Low Word
    MEABUS + 13 = DBUS<7:0>;
                              ! Stored
    IITACKN = 10
                              ! Asserts DTACKN(Added)
     );
                              ! Execute Pending Assignments
    next;
     T = 25
                               ! Return To Phase 2
                              ! Of Clock Cycle 25
     );
                             ! Execute Impending Assignments
    next;
T = 26;
                               ! Clock Cycle 26
                              ! Execute Assignment
next;
PHI1 = lo;
                              ! Phase 2
```

```
PHI2 = hi;
                                  ! Of Clock Cycle 26
SA7 = SA7 - 2;
                                  ! Set System Stack Fointer
                                  ! To Point To Saved Address
                                  ! Causing Exception (Low Word)
                                  ! Execute Pending Assignments
next;
T = 27;
                                   ! Clock Cycle 27
next:
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1
                                  ! Of Clock Cycle 27
PHI2 = 10;
                                  ! Execute Pending Assignments
next:
FHI1 = 10;
                                  ! Phase 2
                                  ! Of Clock Cycle 27
PHI2 = hi;
ASN = hi;
                                  ! Deactivate Address Strobe
LDSN = hi;
                                  ! Deactivate Lower Data Strobe
UDSN = hi;
                                  ! Deactivate Upper Data Strobe
DTACKN = h1:
                                  ! Deactivate Data Transfer(Added)
                                  ! Acknowledge
next;
                                  ! Execute Pending Assignments
T = 28:
                                  ! Clock Cycle 28
next;
                                  ! Execute Pending Assignments
PHI1 = hi;
                                  ! Phase 1 Of
FHI2 = 10;
                                  ! Clock Cycle 28
RW = hi:
                                  ! Memory Read
ADENABLE = 10:
                                  ! Disable Address Rus Buffer
DISENABLE = 10;
                                  ! Disable Data Bus Buffer
IABUS = SA7:
                                  ! Place SA7 On Internal Address
                                  ! Execute Pending Assignments
next;
PHI1 = 10;
                                  ! Phase 2 Of
PHI2 = hi:
                                  ! Clock Cycle 28
ADENABLE = hi;
                                  ! Enable Address Bus Buffer
                                  ! Data Rus High Impedanced
IvBUS = 0xffff;
                                  ! Gate Internal Address Bus
EXABUF = IABUS;
                                  ! Into External Address Buffer
FCMODE = SRMODE:
                                  ! Supervisor Mode
FCSPACE = 1;
                                  ! Accessing Data
                                  ! Place Low Word Of TEMPADR
IDBUS = TEMPADRLOW;
                                  ! Onto Internal Data Bus
                                  ! Execute Impending Assignments
next;
ABUS = EXABUF;
                                  ! Address Placed On Bus(Added)
next;
                                  ! Execute Pending Assignments
T = 29;
                                  ! Clock Cycle 29
next;
                                  ! Execute Assignment
```

```
! Phase 1 Of
PHI1 = hi;
                                  ! Clock Cycle 29
PHI2 = 10:
                                  ! Assert Address Strobe
ASN = 10;
KW = 10:
                                  ! Activate Write
EXDBUF = IDBUS;
                                  ! Internal Data Bus Moved
                                  ! To External Data Ruffer
                                  ! Execute Pending Assignments
next;
FHI1 = 10;
                                  ! Phase 2
                                  ! Of Clock Cycle 29
PHI2 = hi;
DRUS = EXDRUF;
                                  ! Contents Of External Data
                                  ! Buffer Placed On Data Bus
                                  ! Enable Data Bus
DBENABLE = hi;
next;
                                  ! Execute Pending Assignments
T = 30:
                                  ! Clock Cycle 30
next;
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1
PHI2 = 10;
                                  ! Of Clock Cycle 30
UDSN = 10;
                                  ! Activate Upper Data Strobe
                                  ! Activate Lower Data Strobe
LDSN = 10;
twait = 0;
next;
                                  ! Wait For Memory To Place
while ITACKN eql hi
                                  ! Itata On The Bus
     twait = twait + 1;
     next;
                                  ! Execute Impending Assignments
     PHI1 = 10;
                                  ! Phase 2
                                  ! Of Clock Cycle 30
     PHI2 = hi;
     next;
                                  ! Execute Assignments
     T = 31;
                                  ! Clock Cycle 31
                                  ! Execute Assignment
     next;
                                  ! Phase 1
     PHI1 = hi;
                                  ! Of Clock Cycle 31
     FHI2 = 10;
     if twait eql 2
                                 ! TEMPADE LOW WORD
     MEARUS] = DRUS<15:8>;
     MEABUS + 13 = DBUS<7:0>;
                                  ! Stored
                                 ! Asserts DTACKN(Added)
     TITACKN = 10
     );
                                 ! Execute Pending Assignments
     next;
     /*******************************
     T = 30
                                  ! Return To Phase 2
                                  ! Of Clock Cycle 30
     );
```

見れられるとのは最かなならないは、自然ののないなど、自然ないないには、

(

```
next;
                                 ! Execute Impending Assignments
T = 31;
                                  ! Clock Cycle 31
next:
                                 ! Execute Assignment
PHI1 = lo;
                                 ! Phase 2
PHI2 = hi;
                                 ! Of Clock Cycle 31
SA7 = SA7 - 4;
                                 ! Set System Stack Pointer
                                 ! To Point To Saved Access Type
next;
                                 ! Execute Pending Assignments
T = 32;
                                  ! Clock Cycle 32
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
PHI2 = 10:
                                 ! Of Clock Cycle 32
next;
                                 ! Execute Pending Assignments
FHI1 = lo;
                                 ! Phase 2
                                 ! Of Clock Cycle 32
PHI2 = hi;
ASN = hi:
                                 ! Neactivate Address Strobe
LDSN = hi:
                                 ! Deactivate Lower Data Strobe
UDSN = hi;
                                 ! Deactivate Upper Data Strobe
DTACKN = hi;
                                 ! Deactivate Data Transfer(Added)
                                 ! Acknowledge
                                 ! Execute Pending Assignments
next;
T = 33;
                                 ! Clock Cycle 33
next;
                                 ! Execute Pending Assignments
                                 ! Phase 1 Of
PHI1 = hi;
                                 ! Clock Cycle 33
FHI2 = 10:
RW = hi;
                                 ! Memory Read
ADENABLE = 10;
                                 ! Disable Address Bus Ruffer
DRENABLE = 10;
                                 ! Disable Data Bus Buffer
IABUS = SA7:
                                 ! Flace SA7 On Internal Address
                                 ! Bus
                                 ! Execute Pending Assignments
next;
PHI1 = lo;
                                 ! Phase 2 Of
PHI2 = hi:
                                 ! Clock Cycle 33
AUENABLE = hi:
                                 ! Enable Address Bus Buffer
DBUS = 0::ffff;
                                 ! Nata Bus High Impedanced
                                 ! Gate Internal Address Bus
EXABUF = IABUS;
                                 ! Into External Address Buffer
FCMODE = SRMODE;
                                 ! Supervisor Mode
                                 ! Accessing Data
FCSPACE = 1;
IDRUS = ACTYPE;
                                 ! Place ACTYPE
                                 ! Onto Internal Data Bus
                                 ! Execute Impending Assignments
next;
```

```
ABUS = EXABUF;
                                ! Address Flaced On Bus(Added)
                                ! Execute fending Assignments
next;
T = 34;
                                 ! Clock Cycle 34
next;
                                 ! Execute Assignment
                                ! Phase 1 Of
PHI1 = hi;
                                ! Clock Cycle 34
PHI2 = 10:
                                ! Assert Address Strobe
ASN = lo;
                                ! Activate Write
KW = lo;
EXDBUF = IDBUS;
                                ! Internal Data Rus Moved
                                ! To External Data Buffer
                                ! Execute Pending Assignments
next;
FHI1 = 10;
                                ! Phase 2
                                ! Of Clock Cycle 34
PHI2 = hi;
                                ! Contents Of External Data
DRUS = EXDRUF;
                                ! Buffer Placed On Data Bus
                                ! Enable Data Bus
DRENABLE = hi;
                                ! Execute Fending Assignments
next;
! Clock Cycle 35
next;
                                 ! Execute Assignment
PH11 = hi;
                                ! Phase 1
                                 ! Of Clock Cycle 35
PHI2 = 10;
                                 ! Activate Upper Nata Strobe
UUSN = lo;
                                 ! Activate Lower Data Strobe
LIISN = 10;
twalt = 0;
next:
                                 ! Wait For Memory To Place
while DTACKN eql hi
                                 ! Data On The Bus
     twait = twait + 1;
     next;
                                 ! Execute Impending Assignments
                                 ! Phase 2
     fHII = lo;
                                 ! Of Clock Cycle 35
     PHI2 = hi;
     next;
                                 ! Execute Assignments
     ! Clock Cycle 36
     T = 36;
                                ! Execute Assignment
     next;
     PHI1 = hi;
                                 ! Phase 1
     FHI2 = 10;
                                 ! Of Clock Cycle 36
     if twait eql 2
     MEABUS3 = DBUS<15:8>;
                                ! ACTYPE
     MEABUS + 13 = DBUS<7:0>;
                                 ! Stored
     DITACKN = 10
                                ! Asserts DTACKN(Added)
     );
```

```
next;
                               ! Execute Pending Assignments
     ! Return To Phase 2
                               ! Of Clock Cycle 35
     );
     next;
                               ! Execute Impending Assignments
! Clock Cycle 36
T = 36;
                               ! Execute Assignment
next;
                               ! Phase 2
FHI1 = lo;
                               ! Of Clock Cycle 36
PHI2 = hi;
SA7 = SA7 + 2;
                               ! Set System Stack Fointer
                               ! To Point To Address Causing
                               ! Exception (High Word)
next;
                               ! Execute Pending Assignments
! Clock Cycle 37
T = 37:
                               ! Execute Assignment
next;
                               ! Phase 1
PHI1 = hi;
PHI2 = 10;
                               ! Of Clock Cycle 37
                               ! Execute Pending Assignments
next;
PHI1 = lo;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 37
ASN = hi;
                               ! Neactivate Address Strobe
LDSN = hi;
                               ! Deactivate Lower Data Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
DITACKN = hi;
                               ! Deactivate Data Transfer(Added)
                               ! Acknowledge
                               ! Execute Pending Assignments
next:
T = 38;
                               ! Clock Cycle 38
                               ! Execute Pending Assignments
next;
                               ! Phase 1 Of
PH11 = h1;
PHI2 = 10:
                               ! Clock Cycle 38
RW = hi;
                               ! Memory Read
MDENABLE = 10;
                               ! Disable Address Bus Buffer
                               ! Disable Data Bus Buffer
IBENABLE = 10;
lakus = SA7;
                               ! Place SA7 On Internal Address
                               ! Execute Pending Assignments
next;
PHI1 = 10;
                               ! Phase 2 Of
                               ! Clock Cycle 38
THI2 = hi:
                               ! Enable Address Bus Buffer
ADENABLE = hi;
DBUS = 0xffff;
                              . ! Data Bus High Impedanced
```

```
! Gate Internal Address Bus
EXABUF = IABUS;
                                 ! Into External Address Buffer
                                ! Supervisor Mode
FCMODE = SRMODE;
FCSPACE = 1;
                                ! Accessing Data
                               ! Place High Word Of TEMPADR
IDBUS = TEMPADRHI;
                                ! Onto Internal Data Bus
                                 ! Execute Impending Assignments
next;
                                ! Address Placed On Bus(Added)
ARUS = EXABUF;
                                 ! Execute Pending Assignments
next;
! Clock Cycle 39
T = 39;
                                 ! Execute Assignment
next;
                                ! Phase 1 Of
PHI1 = hi;
                                 ! Clock Cycle 39
FHI2 = 10;
                                 ! Assert Address Strobe
ASN = lo;
RW = 10:
                                 ! Activate Write
                                ! Internal Data Bus Moved
EXDBUF = IDBUS;
                                ! To External Data Buffer
                                ! Execute Pending Assignments
next;
PHI1 = lo;
                                ! Phase 2
                                ! Of Clock Cycle 39
PHI2 = hi;
DBUS = EXDBUF;
                                ! Contents Of External Data
                                ! Buffer Placed On Data Bus
                                 ! Enable Data Bus
DBENABLE = hi;
                                 ! Execute Pending Assignments
next;
! Clock Cycle 40
T = 40:
                                 ! Execute Assignment
next;
                                ! Phase 1
PH11 = hi;
                                 ! Of Clock Cycle 40
PHI2 = 10;
                                 ! Activate Upper Data Strobe
UI(SN = 10)
                                 ! Activate Lower Data Strobe
LDSN = 10;
twait = 0;
next:
                                ! Wait For Memory To Place
while ITACKN eql hi
                                ! Data On The Bus
     twait = twait + 1;
                                ! Execute Impending Assignments
     next:
                                 ! Phase 2
     PHI1 = 10;
                                 ! Of Clock Cycle 40
     PHI2 = hi;
                                 ! Execute Assignments
     next;
     T = 41:
                                 ! Clock Cycle 41
     next;
                                 ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
```

```
PHI2 = 10;
                              ! Of Clock Cycle 41
     if twait eql 2
     m[ABUS] = DBUS(15:8):
                              ! TEMPADE High Word
     MEABUS + 13 = DBUS<7:0>;
                               ! Stored
    INTACKN = 10
                              ! Asserts DTACKN(Added)
    );
    next;
                               ! Execute Fending Assignments
    T = 40
                               ! Return To Phase 2
                               ! Of Clock Cycle 40
    );
     next;
                              ! Execute Impending Assignments
T = 41;
                               ! Clock Cycle 41
next;
                               ! Execute Assignment
PHI1 = 10;
                               ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 41
next;
                               ! Execute Pending Assignments
T = 42;
                                 ! Clock Cycle 42
next;
                               ! Execute Assignment
PHI1 = hi;
                               ! Phase 1
                               ! Of Clock Cycle 42
PHI2 = 10;
next;
                               ! Execute Pending Assignments
PHI1 = lo;
                              ! Phase 2
PHI2 = hi;
                               ! Of Clock Cycle 42
ASN = hi:
                               ! Deactivate Address Strobe
LISN = hi;
                               ! Deactivate Lower Data Strobe
UDSN = hi;
                              ! Deactivate Upper Data Strobe
                               ! Deactivate Data Transfer(Added)
DTACKN = hi;
                               ! Acknowledge
next;
                               ! Execute Pending Assignments
T = 43;
                                ! Clock Cycle 43
                               ! Execute Assignment
next;
FHI1 = his
                               ! Phase 1
PHI2 = 10;
                               ! Of Clock Cycle 43
RW = hi
                               ! Memory Read
ADENABLE = 10;
                               ! Disable Address Bus Buffer
DRENABLE = lo;
                               ! Disable Data Bus kuffer
IABUS = VECADR:
                               ! Place VECADR On Internal Address
                               ! Rus
next;
                               ! Execute Pending Assignments
```

```
FHII = 10;
                                 ! Phase 2 Of
PHI2 = hi:
                                 ! Clock Cycle 43
ADENABLE = hi:
                                 ! Enable Address Rus Ruffer
DBUS = 0xffff;
                                 ! Data Bus High Impedanced
EXABUF = IABUS;
                                 ! Gate Internal Address Rus
                                 ! Into External Address Buffer
FCMODE = SRMODE;
                                 ! Supervisor Mode
FCSPACE = 1;
                                 ! Accessing Data
                                 ! Execute Impending Assignments
next;
ABUS = EXABUF;
                                 ! Address Flaced On Eus(Added)
next:
                                 ! Execute Pending Assignments
! Clock Cycle 44
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1 Of
PHI2 = 10;
                                 ! Clock Cycle 44
ASN = lo;
                                 ! Assert Address Strobe
LDSN = lo;
                                 ! Assert Lower Data Strobe
UDSN = lo:
                                 ! Assert Upper Data Strobe
IBENABLE = hi:
                                 ! Enable Data Bus
                                 ! Execute Pending Assignments
next:
FHI1 = lo;
                                 ! Phase 2
                                 ! Of Clock Cycle 44
PHI2 = hi:
                                 ! Execute Fending Assignments
next;
T = 45;
                                 ! Clock Cycle 45
next;
                                 ! Execute Assignment
PHI1 = hi;
                                 ! Phase 1
PHI2 = 10;
                                 ! Of Clock Cycle 45
while DTACKN eql hi
                                 ! Wait For Memory To Place
                                 ! Data On The Bus
     next;
                                 ! Execute Impending Assignments
     FHI1 = lo;
                                 ! Phase 2
     PHI2 = hi;
                                 ! Of Clock Cycle 45
     next:
                                 ! Execute Assignments
     T = 46;
                                 ! Clock Cycle 46
     next;
                                 ! Execute Assignment
     PHI1 = hi;
                                 ! Phase 1
     FHI2 = 10;
                                 ! Of Clock Cycle 46
     DBUS<15:8> = MCABUS3;
                                 ! Memory Places address
     DBUS<7:0> = MEABUS + 13;
                                 ! On Data Bus And
     DTACKN = 10;
                                 ! Asserts DTACKN(Added)
     next;
                                 ! Execute Pending Assignments
```

```
T = 45;
                                 ! Return To Phase 2
                                ! Of Clock Cycle 45
     );
                               ! Execute Impending Assignments
     next;
T = 46;
                                ! Clock Cycle 46
                                ! Execute Assignment
next;
FHI1 = 10;
                                ! Phase 2
                                ! Of Clock Cycle 46
PHI2 = hi;
                                ! Instruction On Nata Bus
EXDRUF = DRUS;
                               ! Is Placed In External Data
                                ! Bus Buffer
next;
                                ! Execute Pending Assignments
! Clock Cycle 47
T = 47:
                               ! Execute Assignment
next;
PH11 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 47
PHI2 = 10;
HANADRLOW = EXDBUF;
                               ! The Contents Of The External
                                ! Data Bus Buffer Are Flaced
                                ! In Handler Routine Low Address
next;
                                ! Execute Pending Assignments
PHI1 = lo;
                                ! Phase 2
                                ! Of Clock Cycle 47
PHI2 = hi;
ASN = hi;
                               ! Deactivate Address Strobe
                               ! Deactivate Lower Data Strobe
LDSN = hi:
                               ! Neactivate Upper Nata Strobe
UDSN = hi;
                               ! Deactivate Data Transfer(Added)
DTACKN = hi;
                                ! Acknowledge
                               ! Increment Vector Address Register
VECADR = VECADR + 2;
                               ! To Pick Handler Address Low Word
                                ! Execute Pending Assignments
next;
T = 48:
                                ! Clock Cycle 48
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1
PHI2 = 10:
                                ! Of Clock Cycle 48
                                ! Memory Read
RW = hi;
                                ! Disable Address Bus Buffer
AUENABLE = 10;
                                ! Disable Data Bus Buffer
DEBENABLE = 10;
                                ! Data Rus High Impedanced
DBUS = 0xffff;
                                ! Place VECADR On Internal Address
IABUS = VECADR;
```

! Rus

```
IDBUS = HANADRLOW;
                                  ! Move Handler High Address To
                                  ! Data Bus
next;
                                  ! Execute Pending Assignments
fHI1 = 10;
                                  ! Phase 2 Of
PHI2 = hi:
                                  ! Clock Cycle 48
ADENABLE = hi;
                                  ! Enable Address Bus Buffer
EXABUF = IABUS;
                                  ! Gate Internal Address Bus
                                  ! Into External Address Buffer
FCMODE = SRMODE:
                                  ! Supervisor Mode
FCSPACE = 1;
                                  ! Accessing Data
                                  ! Move Handler High Address
HANADRHI = IDBUS;
                                  ! To Upper Word Of Register
                                  ! Execute Impending Assignments
next;
ABUS = EXABUF;
                                  ! Address Placed On Rus(Added)
next;
                                  ! Execute Pending Assignments
T = 49;
                                  ! Clock Cycle 49
next;
                                  ! Execute Assignment
                                  ! Phase 1 Of
PHI1 = hi:
fHI2 = 10;
                                  ! Clock Cycle 49
ASN = lo;
                                  ! Assert Address Strobe
LDSN = lo;
                                  ! Assert Lower Data Strobe
UDSN = lo:
                                  ! Assert Upper Data Strobe
DEENABLE = hi;
                                  ! Enable Data Rus
'next;
                                  ! Execute Pending Assignments
FHII = lo:
                                  ! Phase 2
PHI2 = hi;
                                  ! Of Clock Cycle 49
next;
                                  ! Execute Pending Assignments
T = 50:
                                  ! Clock Cycle 50
next;
                                  ! Execute Assignment
PHI1 = hi:
                                  ! Phase 1
                                  ! Of Clock Cycle 50
PHI2 = 10;
while DTACKN eql hi
                                 ! Wait For Memory To Place.
                                  ! Data On The Bus
     next;
                                  ! Execute Impending Assignments
     PHI1 = 10;
                                  ! Phase 2
     PHI2 = hi;
                                  ! Of Clock Cycle 50
                                  ! Execute Assignments
     next;
      ' Clock Cycle 51
     T = 51;
     next;
                                  ! Execute Assignment
     PHI1 = hi;
                                  ! Phase 1
                                  ! Of Clock Cycle 51
     PH12 = 10:
```

```
! Memory Flaces Address
     DBUS<15:8> = MCABUS];
                               ! On Data Rus And
    I/BUS<7:0> = M[ABUS + 1];
    DTACKN = 10:
                                ! Asserts DTACKN(Added)
    next:
                                ! Execute Pending Assignments
     T = 50;
                                 ! Return To Phase 2
                               ! Of Clock Cycle 50
     );
                               ! Execute Impending Assignments
     next:
T = 51:
                                ! Clock Cycle 51
                                ! Execute Assignment
next;
                               ! Phase 2
f'HI1 = lo;
PHI2 = hi;
                                ! Of Clock Cycle 51
EXDBUF = DBUS;
                               ! Instruction On Data Bus
                                ! Is Placed In External Data
                                ! Bus Buffer
                               ! Execute Pending Assignments
next;
! Clock Cycle 52
T = 52:
                                ! Execute Assignment
next;
PH11 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 52
PHI2 = 10;
HANADRLOW = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                                ! In Handler Routine Low Address
next;
                                ! Execute Pending Assignments
PHI1 = lo;
                               ! Phase 2
                               ! Of Clock Cycle 52
PHI2 = hi;
ASN = hi:
                               ! Deactivate Address Strobe
LUSN = hi;
                               ! Deactivate Lower Data Strobe
                                ! Deactivate Upper Data Strobe
UDSN = hi;
DTACKN = hi;
                                ! Deactivate Data Transfer(Added)
                                ! Acknowledge
next;
                                ! Execute Pending Assignments
T = 53;
                                 ! Clock Cycle 53
next;
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1 Of
PHI2 = 10;
                                ! Clock Cycle 53
                                ! Memory Read
RW = hi;
ADENABLE = lo;
                                ! Disable Address Bus Buffer
                                ! Disable Data Bus Buffer
DIBENABLE = 10;
INBUS = 0xffff;
                                ! Data Rus High Impedanced
```

```
IABUS = HANADR;
                                 ! Place HANADR On Internal Address
                                 ! Rus
next;
                                ! Execute Pending Assignments
PHI1 = lo:
                                ! Phase 2 Of
PHI2 = hi;
                                ! Clock Cycle 53
                                ! Enable Address Bus Buffer
ADENABLE = hi:
EXABUF = IABUS;
                                ! Gate Internal Address Bus
                                ! Into External Address Buffer
FCMODE = SRMODE;
                                ! User Mode
PC = TABUS:
                                ! Place HANADR in PC
FCSPACE = 2;
                                ! Accessing Program
next;
                                ! Execute Impending Assignments
                                ! Address Placed On Bus(Added)
ABUS = EXABUF;
                                ! Execute Pending Assignments
next:
T = 54:
                                 ! Clock Cycle 54
next;
                                 ! Execute Assignment
FHI1 = hi;
                                 ! Phase 1 Of
                                ! Clock Cycle 54
PHI2 = 10:
ASN = 10;
                                ! Assert Address Strobe
LDSN = lo;
                                ! Assert Lower Data Strobe
LITISN = lo:
                                ! Assert Upper Data Strobe
DRENABLE = hi;
                                ! Enable Data Bus
next:
                                ! Execute Pending Assignments
PHI1 = lo;
                                ! Phase 2
PHI2 = hi;
                                ! Of Clock Cycle 54
next;
                                 ! Execute Pending Assignments
T = 55:
                                 ! Clock Cycle 55
next:
                                 ! Execute Assignment
PHI1 = hi:
                                ! Phase 1
PHI2 = 10;
                                ! Of Clock Cycle 55
while DTACKN eql hi
                                ! Wait For Memory To Place
                                ! Data On The Bus
     next;
                                ! Execute Impending Assignments
     PHI1 = 10;
                                ! Phase 2
     PHI2 = hi;
                                ! Of Clock Cycle 55
     next:
                                ! Execute Assignments
     T = 56:
                                ! Clock Cycle 56
     next;
                                 ! Execute Assignment
     PHI1 = hi;
                                ! Phase 1
     PHI2 = 10;
                                ! Of Clock Cycle 56
     DBUS<15:8> = MEABUS3;
                                ! Memory Places Instruction
```

```
DBUS<7:0> = MEABUS + 13;
                              ! On Data Bus And
    DTACKN = lo;
                               ! Asserts DTACKN(Added)
    next;
                               ! Execute Pending Assignments
    T = 55:
                                ! Return To Phase 2
                               ! Of Clock Cycle 55
    );
                              ! Execute Impending Assignments
    next;
T = 56;
                                ! Clock Cycle 56
next;
                               ! Execute Assignment
FHI1 = lo:
                               ! Phase 2
PHI2 = hi:
                               ! Of Clock Cycle 56
EXDBUF = DBUS;
                               ! Instruction On Data Rus
                               ! Is Placed In External Data
                               ! Bus Buffer
next;
                               ! Execute Pending Assignments
T = 57;
                                ! Clock Cycle 57
next:
                               ! Execute Assignment
PHI1 = hi:
                               ! Phase 1
PHI2 = 10:
                               ! Of Clock Cycle 57
PFR = EXDBUF;
                               ! The Contents Of The External
                               ! Data Bus Buffer Are Placed
                               ! In Prefetch Register
next;
                               ! Execute Pending Assignments
PHI1 = 10;
                               ! Phase 2
                               ! Of Clock Cycle 57
PHI2 = hi:
                               ! Deactivate Address Strobe
ASN = hi:
LDSN = hi;
                               ! Neactivate Lower Nata Strobe
UDSN = hi;
                               ! Deactivate Upper Data Strobe
IR = PFR;
                               ! Contents Of Prefetch Register
                               ! Are Placed Into Instruction
                               ! Register
ITACKN = hi;
                               ! Deactivate Data Transfer(Added)
                               ! Acknowledge
PC = PC + 2;
                               ! Increment Program Counter
next;
                               ! Execute Pending Assignments
T = 58;
                               ! Clock Cycle 58
next:
PHI1 = hi;
                               ! Phase 1 Of
PHI2 = 10;
                               ! Clock Cycle 58
ADENABLE = 10;
                               ! Disable Address Bus Buffer
```

```
DRENABLE = 10;
                                          ! Disable Data Bus Buffer
     DRUS = 0xffff;
                                          ! Data Bus High Impedanced
     ASN = hi;
                                          ! Disable Address,
     LDSN = hi;
                                          ! Lower Data, and
     UDSN = hi;
                                          ! Upper Data Strobes
                                          ! Execute Pending Assignments
     next:
     PHI1 = lo;
                                          ! Phase 2 Of
     PHI2 = hi:
                                          ! Clock Cycle 58
     next;
                                          ! Execute Impending Assignments
     ! Clock Cycle 59
     next;
                                          ! Execute Assignment
                                          ! Phase 1 Of
     PHI1 = hi:
     PHI2 = lo;
                                          ! Clock Cycle 59
     next;
                                          ! Execute Pending Assignments
     FHI1 = lo:
                                          ! Phase 2
     PHI2 = hi;
                                          ! Of Clock Cycle 59
     next;
     T = 0
                                          ! Reset Clock Cycle Counter
rte :=
                                           ! RTE (Return From Exception)
   SA7 = SA7 - 2;
                                           ! Effect Of This Instruction
                                           ! Is To Pop The PC And SR
   next;
   SR<15:8> = MESA73;
                                           ! From The Stack
   SR<7:0> = MCSA7 + 13;
   next;
   SA7 = SA7 + 2;
   next;
   PC<31:24> = MESA73;
   PC<23:16> = MESA7 + 13;
   next;
   SA7 = SA7 + 2;
   next:
   PC<15:8> = MESA73;
   PC<710> = MESA7 + 13;
   next:
   IR<15:8> = MCPC3;
   IR<7:0> = MIPC + 13;
   next:
   FC = PC + 2;
   next:
                                           ! Requires 20 Clock Cycles
   T = 19;
   next;
   T = 0
```

```
move :=
                                       ! MOVE.W DI,(A1)
     PHI1 = hi:
                                       ! Phase 1 Of
     PHI2 = 10;
                                       ! Clock Cycle 0
     DBUS = 0xffff;
                                       ! Data Bus High Impedanced
     KW = hi:
                                      ! Memory Read
     ADENABLE = 10;
                                      ! Disable Address Bus Buffer
     DIBENABLE = 10;
                                       ! Disable Data Bus Buffer
     InBUS = PC;
                                      ! Place PC On Internal Address
                                       ! Bus
     next;
                                       ! Execute Pending Assignments
     PHI1 = 10;
                                      ! Phase 2 Of
     PHI2 = hi;
                                      ! Clock Cycle O
     ADENABLE = hi;
                                      ! Enable Address Rus Ruffer
     EXABUF = IABUS;
                                      ! Gate Internal Address Bus
                                       ! Into External Address Buffer
     FCMODE = SRMODE;
                                      ! User Mode
     FCSPACE = 2;
                                      ! Accessing Program
     next:
                                      ! Execute Impending Assignments
     ABUS = EXABUF;
                                       ! Address Placed On Bus(Added)
     next;
                                       ! Execute Pending Assignments
     T = 1;
                                       ! Clock Cycle 1
     next;
                                       ! Execute Assignment
     FHII = hi
                                       ! Phase 1 Of
     PHI2 = 10;
                                       ! Clock Cycle 1
     ASN = 10;
                                       ! Assert Address Strobe
     LISN = lo;
                                       ! Assert Lower Data Strobe
     UDSN = lo;
                                       ! Assert Upper Data Strobe
     DBENABLE = hi;
                                       ! Enable Data Rus
     next:
                                       ! Execute Pending Assignments
     PHI1 = 10;
                                       ! Phase 2
     PHI2 = hi;
                                       ! Of Clock Cycle 1
     next;
                                       ! Execute Pending Assignments
     T = 2;
                                      ! Clock Cycle 2
     next;
                                       ! Execute Assignment
     PHI1 = hi;
                                      ! Phase 1
     PHI2 = 10;
                                      ! Of Clock Cycle 2
                                      ! Wait For Memory To Place
     while DTACKN eql hi
                                      ! Data On The Bus
                                      ! Execute Impending Assignments
          next;
          PHI1 = lo;
                                      ! Phase 2
```

```
PHI2 = hi:
                                ! Of Clock Cycle 2
    next:
                                ! Execute Assignments
     T = 3:
                                ! Clock Cycle 3
                                ! Execute Assignment
    next;
    PHI1 = hi;
                                ! Phase 1
                                ! Of Clock Cycle 3
    PHI2 = 10;
     DBUS<15:8> = MCABUS3;
                               ! Memory Places Instruction
    IGBUS<7:0> = MIABUS + 13;
                               ! On Data Bus And
     DITACKN = lo;
                                ! Asserts DTACKN(Added)
    next:
                                ! Execute Pending Assignments
     T = 2
                                ! Return To Phase 2
                                ! Of Clock Cycle 2
     );
    next;
                                ! Execute Impending Assignments
T = 3;
                                ! Clock Cycle 3
                                ! Execute Assignment
next;
                                ! Phase 2
PHI1 = lo;
                                ! Of Clock Cycle 3
PHI2 = hi;
EXDRUF = DRUS;
                                ! Instruction On Nata Bus
                                ! Is Placed In External Data
                                ! Bus Buffer
                                ! Execute Pending Assignments
next;
T = 4;
                                ! Clock Cycle 4
nexti
                                ! Execute Assignment
PHI1 = hi;
                                ! Phase 1
FHI2 = 10;
                                ! Of Clock Cycle 4
PFR = EXDBUF;
                               !! The Contents Of The External
                                ! Data Bus Buffer Are Placed
                                ! In Prefetch Register
                                ! Execute Pending Assignments
next:
                                ! Phase 2
PHI1 = lo;
                                ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi:
                                ! Deactivate Address Strobe
LUSN = hi;
                                ! Deactivate Lower Data Strobe
UDISN = h1;
                                ! Deactivate Upper Data Strobe
                                ! Are Flaced Into Instruction
                                ! Register
PC = PC + 2;
                                ! Increment Program Counter
DITACKN = hi;
                                ! Neactivate Nata Transfer(Added)
                                ! Acknowledge
next;
```

```
T = 5;
                                   ! Clock Cycle 5
next;
                                   ! Execute Previous Assignment
PHI1 = ni;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 5
RW = hi;
                                   ! Memory Read
ADENABLE = 10;
                                   ! Disable Address Bus Buffer
DRUS = Oxffff;
                                   ! Data Bus Returned To High
                                   ! Impedance State
                                   ! Disable Data Bus Buffer
DIRENABLE = 10;
                                   ! Place A[1] On Internal Address
IABUS = AC13;
                                   ! Bus
                                   ! Execute fending Assignments
next;
PHI1 = lo;
                                   ! Phase 2 Of
                                   ! Clock Cycle 5
PHI2 = hi;
                                   ! Enable Addréss Bus Buffer
ADENABLE = hi;
FCMODE = SRMODE;
                                   ! User Mode
FCSPACE = 1;
                                   ! Accessing Program
EXABUF = IABUS;
                                   ! Gate Internal Address Bus
IDBUS = D1LWORD;
                                   ! Place Low Word from D[1] On
                                   ! Internal Data Rus
next;
                                   ! Into External Address Buffer
ABUS = EXABUF;
                                   ! Address Placed On Bus(Added)
                                   ! Execute Pending Assignments
! Clock Cycle 6
next;
                                    ! Execute Assignment
PHI1 = hi;
                                   ! Phase 1 Of
PHI2 = 10;
                                   ! Clock Cycle 6
ASN = lo;
                                   ! Assert Address Strobe
KW = 10;
EXDBUF = IDBUS;
                                   ! Place Contents Of Internal
                                    ! Data Bus Into External Data Buffer
SRCARRY = 10;
                                    ! Reset Condition Code Bits
SROVER = lo;
SRZERO = lo;
SENEG = 10:
next;
                                   ! Execute Fending Assignments
                                   ! Phase 2
PHI1 = 10;
PHI2 = hi;
                                    ! Of Clock Cycle 6
DBUS = EXDBUF;
                                   ! Place Data On External Data Bus
                                   ! Enable Data Bus
DBENABLE = hi;
                                    ! This Instruction Will Abort
case ABUS(1>
     0:
if EXDRUF eq1 0
                                   ! Set Zero Condition Bit If Needed
```

■特なななられる■■なれるののの■■ないなったのは

SKZERO = hi:

```
next;
                             ! Execute Pending Assignments
T = 7;
                             ! Clock Cycle 7
next;
                             ! Execute Assignment
FHI1 = hi;
                             ! Phase 1
PHI2 = 10;
                             ! Of Clock Cycle 7
if EXDBUF<15>
                             ! Set Negative Condition Bit
                             ! If Needed
  SRNEG = hi;
UDSN = lo;
                             ! Activate Upper And
                             ! Lower Data Strobes
LUSN = lo;
twait = 0:
                             ! Wait Cycle Counter Initialized
next;
while DTACKN eql hi
                             ! Wait For Memory To Place
                             ! Data On The Bus
    twait = twait + 1;
                             ! Increment Wait Cycle
    next:
                             ! Execute Impending Assignments
    PHI1 = lo:
                             ! Phase 2
                             ! Of Clock Cycle 7
    PHI2 = hi;
    next:
                             ! Execute Assignments
    T = 8;
                             ! Clock Cycle 8
    next;
                             ! Execute Assignment
    PHI1 = hi;
                             ! Phase 1
                             ! Of Clock Cycle 8
    PHI2 = 10;
    if twait eql 2
                            ! Memory Responds After 2 Cycles
    MIABUS3 = DBUS<15:8>:
                           ! Store Data From Bus
    MEABUS + 13 = DBUS<7:0>;
                             ! In Memory
                            ! Asserts DTACKN(Added)
    DITACKN = 10
    );
    next;
                            ! Execute Pending Assignments
    T = 7
                            ! Return To Phase 2
                            ! Of Clock Cycle 7
    );
    next;
                            ! Execute Impending Assignments
T = 8:
                             ! Clock Cycle B
next;
                             ! Execute Assignment
PHI1 = 10:
                             ! Phase 2
PHI2 = hi:
                             ! Of Clock Cycle 8
next:
                             ! Execute Pending Assignments
T = 9:
                             ! Clock Cycle 9
```

```
! Execute Assignment
next;
PHI1 = hi;
                                        ! Phase 1
                                        ! Of Clock Cycle 9
PHI2 = 10;
next;
                                        ! Execute Pending Assignments
                                        ! Phase 2
PHI1 = lo;
                                        ! Of Clock Cycle 9
PHI2 = hi;
                                        ! Deactivate Address Strobe
ASN = hi;
                                        ! Deactivate Lower Data Strobe
LDSN = hi;
                                        ! Deactivate Upper Data Strobe
UDSN = hi;
PC = PC + 2;
                                        ! Increment Program Counter
                                        ! Place Contents Of Prefetch
IR = PFR;
                                        ! Register Into Instruction
                                        ! Register
                                        ! Deactivate Data Transfer
DTACKN = hi:
                                        ! Acknowledge(Added)
                                       ! Execute Pending Assignments
next
  )
1:
                                          ! Terminate MOVE.W D1,(A1)
  (
                                        ! Instruction Because Of Illegal
next;
                                        ! Address And Initiate Exception
T = 7;
                                        ! Processing
next;
PHI1 = hi;
PHI2 = 10;
UDSN = lo;
LUSN = lo;
next;
PHI1 = lo;
PHI2 = hi;
                                       ! Instruction Register And
IRTEMP = IK;
                                       ! User/System Data/Program
ACTYPE<2:0> = FC;
                                       ! Status Saved In Temporary Registers
EXCEPT = hi;
                                       ! Exception Occurred During
ACTYPE<4> = RW;
                                       ! Write Cycle
next;
T = 8;
                                         ! Clock Cycle 8
next;
PHI1 = hi;
PHI2 = 10;
                                    ! Instruction Caused Exception(Changed)
ACTYPE<3> = 10;
next;
PHI1 = 10;
PHI2 = hi;
ASN = hi;
UDSN = hi;
                                    ! Illegal Address Exception
IR = 0 \times 0 \text{ of } d;
```

```
LDSN = h1;
     next
                                       ! Exception Processing
     esac;
     T = 0
=: and
                                        ! JMP (A0)
     PHI1 = hi;
                                        ! Phase 1 Of
     PHI2 = 10:
                                        ! Clock Cycle 0
     DBUS = 0xffff;
                                        ! Place Data Bus In A High
                                        ! Impedance State (Added)
     RW ≈ hi:
                                        ! Memory Read
     ADENABLE = 10;
                                        ! Disable Address Bus Ruffer
     DBENABLE = 10;
                                        ! Disable Data Bus Buffer
     IABUS = PC:
                                        ! Place PC On Internal Address
                                        ! Bus
     next;
                                        ! Execute Pending Assignments
     PHI1 = lo;
                                        ! Phase 2 Of
     PHI2 = hi;
                                        ! Clock Cycle 0
     ADENABLE = hi;
                                        ! Enable Address Bus Buffer
     EXABUF = IABUS;
                                        ! Gate Internal Address Bus
                                        ! Into External Address Buffer
     FCMODE = SRMODE;
                                        ! User Mode
     FCSPACE = 2;
                                        ! Accessing Program
     next;
                                        ! Execute Pending Assignments
     ARUS = EXARUF;
                                        ! Address flaced On Bus(Added)
                                        ! Execute Pending Assignments
     next;
     T = 1;
                                        ! Clock Cycle 1
     next;
                                        ! Execute Assignment
     PHI1 = hi;
                                        ! Phase 1 Of
     PHI2 = 10;
                                        ! Clack Cycle 1
     ASN = lo:
                                        ! Assert Address Strobe
     LISN = 10:
                                        ! Assert Lower Data Strobe
     UIISN = 10;
                                        ! Assert Upper Data Strobe
                                        ! Move Jump Address From A[0]
     IABUS = A[O];
                                        ! To Internal Address Buffer
     DBENABLE = h1;
                                        ! Enable Data Rus
     next;
                                        ! Execute Pending Assignments
     PHI1 = 10;
                                        ! Phase 2
     PHI2 = hi:
                                        ! Of Clock Cycle 1
     PC = IABUS;
                                        ! Place Jump Address Into Program
                                        ! Counter
```

```
next;
                              ! Clock Cycle 2
next;
                              ! Execute Assignment
PHI1 = hi;
                              ! Phase 1
FHI2 = 10;
                              ! Of Clock Cycle 2
                              ! Wait For Memory To Place
while DTACKN eql hi
                              ! Nata On The Rus
    next;
                              ! Execute Impending Assignments
    PH11 = 10;
                              ! Phase 2
    PHI2 = hi;
                              ! Of Clock Cycle 2
    next;
                              ! Execute Assignments
    ! Clock Cycle 3
    next;
                              ! Execute Assignment
    FHI1 = h1;
                              ! Phase 1
                              ! Of Clock Cycle 3
    PHI2 = 16;
    DBUS<15:8> = MCABUS];
                              ! Memory Places Instruction
    DBUS<7:0> = MEABUS + 13;
                              ! On Data Bus And
    IJTACKN = 10;
                              ! Asserts INTACKN(Added)
    next;
                              ! Execute Pending Assignments
    ! Return To Phase 2
    T = 2
                              ! Of Clock Cycle 2
    );
    next;
                              ! Execute Impending Assignments
T = 3:
                              ! Clock Cycle 3
next;
                              ! Execute Assignment
PHI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 3
PHI2 = hi;
EXHBUF = DRUS;
                              ! Instruction On Nata Bus
                              ! Is Placed In External Data
                              ! Bus Buffer
next;
                              ! Execute Pending Assignments
T = 4;
                              ! Clock Cycle 4
                              ! Execute Assignment
next;
PHI1 = hi;
                              ! Phase 1
PHI2 = 10;
                              ! Of Clock Cycle 4
next;
PFR = EXDBUF;
                              ! The Contents Of The External
                              ! Data Bus Buffer Are Placed
```

```
! In Prefetch Register
next;
                                  ! Execute Pending Assignments
FHI1 = lo;
                                  ! Phase 2
                                  ! Of Clock Cycle 4
PHI2 = hi;
ASN = hi;
                                  ! Deactivate Address Strobe
                                  ! Deactivate Lower Data Strobe
LUSN = hi;
                                  ! Deactivate Upper Data Strobe
UIISN = hi;
DTACKN = h1;
                                  ! Deactivate Data Transfer
                                  ! Acknowledge(Added)
T = 5;
                                  ! Clock Cycle 5
next;
                                  ! Execute Previous Assignment
                                  ! Phase 1 Of
PHI1 = hi;
FHI2 = 10;
                                  ! Clock Cycle 5
                                  ! Memory Read
RW = hi;
                                  ! Disable Address Bus Buffer
ADENABLE = lo;
                                  ! Disable Data Bus Buffer
DIKENABLE = 10;
1ABUS = PC;
                                  ! Place PC On Internal Address
                                  ! Execute Pending Assignments
next;
PHI1 = lo;
                                  ! Phase 2 Of
PHI2 = hi;
                                  ! Clock Cycle 5
ADENABLE = hi;
                                  ! Enable Address Bus Buffer
FCMODE = SRMODE;
                                  ! User Mode
FCSFACE = 2;
                                  ! Accessing Program
                                  ! Gate Internal Address Bus
EXABUF = IABUS;
                                  ! Into External Address Buffer
next:
ABUS = EXABUF;
                                  ! Address Placed On Bus(Added)
                                  ! Execute Fending Assignments
next;
T = 6;
                                  ! Clock Cycle 6
next;
                                  ! Execute Assignment
PHI1 = hi;
                                  ! Phase 1 Of
PHI2 = 10;
                                  ! Clock Cycle 6
                                  ! Assert Address Strobe
ASN = lo:
                                  ! Assert Lower Data Strobe
LUSN = lo:
                                  ! Assert Upper Data Strobe
UI(SN = 10;
                                  ! Enable Data Bus
DBENABLE = hi;
                                  ! Execute Pending Assignments
next;
PHI1 = lo;
                                  ! Phase 2
PH12 = hi;
                                  ! Of Clock Cycle 6
                                  ! Execute Pending Assignments
           T = 7;
                                  ! Clock Cycle 7
next;
                                  ! Execute Assignment
```

```
PHI1 = hi:
                              ! Phase 1
FHI2 = 10;
                              ! Of Clock Cycle 7
                              ! Wait For Memory To Place
while DTACKN eql hi
                              ! Data On The Rus
    next;
                              ! Execute Impending Assignments
    FHI1 = lo;
                              ! Phase 2
    PHI2 = hi;
                              ! Of Clock Cycle 7
    next;
                              ! Execute Assignments
    T = 8;
                              ! Clock Cycle 8
    next;
                              ! Execute Assignment
                              ! Phase 1
    PHI1 = hi:
                              ! Of Clock Cycle 8
    FHI2 = 10;
    DBUS<15:8> = MEABUSJ;
                              ! Memory Places Instruction
    DBUS<7:0> = MCABUS + 13;
                              ! On Data Bus And
    IITACKN = 10;
                              ! Asserts ITACKN(Added)
                              ! Execute Pending Assignments
    next;
    T = 7
                              ! Return To Phase 2
                              ! Of Clock Cycle 7
    );
                              ! Execute Impending Assignments
    next;
T = 8;
                              ! Clock Cycle 8
                              ! Execute Assignment
next;
PHI1 = lo;
                              ! Phase 2
                              ! Of Clock Cycle 8
PHI2 = hi;
EXDRUF = DBUS:
                              ! Instruction On Data Bus
                              ! Is Placed In External Data
                              ! Bus Buffer
                              ! Execute Pending Assignments
next;
T = 9;
                              ! Clock Cycle 9
next;
                              ! Execute Assignment
                              ! Phase 1
PHI1 = hi;
                              ! Of Clock Cycle 9
PHI2 = 10:
PFF = EXDBUF;
                              ! The Contents Of The External
                              ! Data Bus Ruffer Are Placed
                              ! In Prefetch Register
                              ! Execute Pending Assignments
next;
PHI1 = 10;
                              ! Phase 2
PH12 = hi;
                              ! Of Clock Cycle 9
ASN = hi;
                              ! Deactivate Address Strobe
```

```
! Neactivate Lower Nata Strobe
     LUSN = hi;
                                            ! Deactivate Upper Data Strobe
     UDSN = hi;
                                            ! Increment Program Counter
     PC = PC + 4;
                                            ! Place Contents Of Prefetch
     IR = PFR;
                                            ! Register Into Instruction
                                            ! Register
                                            ! Deactivate Data Transfer
     DTACKN = hi:
                                            ! Acknowledge(Added)
                                            ! Execute Pending Assignments
     next;
                                            ! Reset Clock Cycle Counter
     T = 0
decode_execute_prefetch :=
                        case IR
                                            ! MOVE.W D1, (A1) with illegal addr
                             0x3281: Nove
                                            ! AND.W ##DFFF,SR
                             0x027c: andi
                                            ! AUDQ.L #4,A7
                             0x588f: addq
                                             ! JMP (A0)
                             047320: Jmp
                                            ! RTE (Return From Exception)
                             0x4e73: rte
                             OxOafd: illegal! Illegal Address Exception
                        esac
                        )
main :=
     power_on_initialize;
     fetch_initial_instruction;
     while READY eql hi
           decode_execute_prefetch
```

Appendix D: MC68000 hetamicro Description

 \mathcal{T}

```
1 *
1* m68000.m.
                                          ж I
!* metaMicro description file for Motorola 68000
!* microprocessor.
                                          * 1
!* This file is included in the first line of
                                          *!
!* assembler source code file. say source.m.
!* It generates output file source.n if "micro" is
!* used. If "mas" is used than it generates the
!* following output files.
!* source.n : nodal output file.
!* source.l : assembler listing, logical addresses.
!* source.L : assembler listing, both logical and
                                          ¥ 1
! *
           physical addresses and assembled
                                          * I
! *
           object code listings.
!* l.out
         : assmbled object code core image.
                                          *!
!* Use "micro" with "cater" and "merge" or use "mas".*!
1 *
                                          *!
!* Author
         : Samir S. Shah.
                                          * 1
!* Date
         : Fall 1979.
!* Modified : Samir S. Shah.
         : Spring 1981.
:* Nate
                                          35. I
! ×
                                          *!
*!
                                          *!
!* Declaration for instruction lengths and widths.
!* Memory is byte addressable, default instruction
                                          *!
!* length is 2 bytes,maximum instruction length is
                                          * 1
!* 10 bytes.
1 *
instr
 IE10,23<8>$
format
*!
* Fields of instruction bytes zero and one.
                                          *!
                                          * 1
Opcode
         = 1003<7:4>,
```

```
Condition = IEO3<3:0>,
 Dst Kn
        = I[0]<3:1>,
 Dst_mode0 = ILO3<0>,
 Aux_Op
        = If 03<0>,
 Dst_Mode1 = IE13<7:6>,
 Size
        = 1013<7:6>,
        = 1013<5:3>,
 Src Mode
 I_{-}E
        = I[1]<5>,
 SE
        = I[13<4>,
        = I[1]<3>,
 B M
        = I[i]<2:0>,
 Src Rn
! *
                                    *!
                                    * !
!* Fields of instruction bytes Two and three.
! *
                                    * 1
T Ind = T[23<7>.
     = I[2]<6:4>,
 T En
 T_{size} = II2J\langle 3 \rangle,
 T Disp = I[3]<7:0>,
*!
1 *
!* Fields of instruction bytes Four and five.
                                    *!
                                    * 1
1 *
F Ind = I[4]<7>,
 F Rn = I[4]<6:4>,
 FSize = I[4]<3>,
 F Disp = 1053 < 7:0 >,
*!
! *
                                    *!
!* Fields of instruction bytes Six and seven.
                                    *!
! *
S Ind = 1063 < 7 >,
 SiRn
     = I[6]<6:4>,
 S_Size = I[6]<3>,
 S_Disp = IE73<7:0>,
*!
! *
                                    *!
!* Shorter names for instruction bytes.
1 🕸
                                    *!
IO = I[0]\langle 7:0 \rangle,
 I1 = I[1]<7:0>,
```

(

```
12 = IC23<7:0>,
 13 = 1033 < 7:0 >,
 I4 = I[4]<7:0>,
 15 = IE53<7:0>,
 16 = I[6]<7:0>,
 17 = 1073<7:0>,
 18 = I[8]<7:0>.
 19 = 1[9]<7:0>$
matro
!* Data registers.
                                     *!
                                     * !
100 = 0.8,
 I(1 = 1 &,
 D2 = 2 &,
 13 = 3 &,
 T14 = 4 8.
 105 = 5 &,
 D6 = 6 &,
 107 = 7 &,
! *
                                     *!
!* Address registers.
                                     * 1
! *
                                     *!
A0 = 0 &,
 A1 = 1 &,
 A2 = 2 %
 A3 = 3 &,
 64 = 4 &,
 A5 = 5 &,
 A6 = 6 &,
 A7 = 7 &,
 SF = 7 %,
! *
                                     *!
!* Size.
                                     *!
 B = 0 &
  = 1 &,
```

```
*!
! *
                                       *!
 Addressing modes.
! *
                                       * 1
! Data Register direct
 DR = 0.8,
 AR = 1 &,
           ! Address Register direct
 IR = 2.8,
           ! Indirect Register
 AI = 3 &,
           ! Auto-Increment
 AD = 4 %,
           ! Auto-Decrement
 IS = 5 %,
            DiSplacement
 1X = 6 &,
           ! IndeX
 SF = 7 %,
           ! SPecial
   SH = 0 %, ! SHort
   LN = 1 %, ! Long
   PD = 2 %, ! Frogram counter Displacement
   FX = 3 %, ! Program counter indeX
   Im = 4 %, ! Immediate
X!
!* Register or Memory mode selection.
                                       *!
B = 0 2,
 M = 1 &,
! *
                                       *1
!* Condition codes.
! *
                                       *!
&, ! True
   ≔ ()
   = 1
       %, ! False
 HI = 2
       &, ! Hlgh
 LS = 3
       &, ! Low or Same
 CC = 4
       &, ! Carry Clear
 CS = 5
       %, ! Carry Set
 NE = 6
       &, ! Not Equal
 EQ = 7
       &, ! EQual
 VC = 8
       &, ! oVerflow Clear
 US = 9
       &, ! oVerflow Set
 PL = 10 &, ! PLus
 MT = 11 &, ! MInus
 GE = 12 %, ! Greater or Equal
 LT = 13 \&, ! Less Than
 GT = 14 %, ! Greater Than
 LE = 15 %, ! Less or Equal
```

```
1 *
!* Addressing mode macros.
                                              *!
! *
sDR(rn) =
   if 'rn eql "^SR$" then
      Src_Mode = 7;
      Src_Rn = 4
   else
      Src_Mode = DR;
    Src_Rn = rn
      3 &,
 sAR(rn) =
   Src_Mode = AR;
   Src_Rn = rn &,
 sIR(rn) =
   Src_Mode = Ik;
   Src_Rn = rn &,
 sAI(rn) =
   Src Mode = Al;
   Src_Rn = rn &,
 5AD (rn) =
   Src_Mode = AD;
   Src_Rn = rn %,
 dDR (rn) =
   Dst_Mode0 = DR ^ -2;
   Dst_Mode1 = DR;
   Dat_En = rn &,
 dAR(rn) =
   Dst_Mode0 = AR \land -2;
   Dst_Mode1 = AR;
   Det_Rn = rn &,
 dIR(rn) =
   Dst_Mode0 = IR ^ -2;
   Dst_Mode1 = IR;
   Dst_Rn = rn &,
 dAI (rn) =
   Iist_Mode0 = AI \land -2;
   Dst_mode1 = AI;
   Dst_Rn = rn &,
```

```
dAD(rn) =
  Dst_Mode0 = AD \land -2;
  Dst_Mode1 = AD;
  Dst_En = rn &,
app_W (addr) =
  if length eql 2 then
     I2 = addr \wedge -8;
     I3 = addr
    >;
  if length eql 4 then
     14 = addr ^ -8;
     15 = addr
  if length eql 6 then
     I6 = addr ^ -8;
     I7 = addr
    3;
  length = length + 2 %,
app_L (addr) =
  if length eql 2 then
     I2 = addr \wedge -24;
     I3 = addr ^ -16;
     I4 = addr \wedge -8;
     15 = addr
    ን ;
  if length eql 4 then
     14 = addr \wedge -24;
     I5 = addr ^ -16;
     16 = addr ^ -8;
     I7 = addr
  if length eql 6 them
     T6 = uddr \wedge -24;
     I7 = addr \wedge -16;
     I8 = addr ^ -8;
     I9 = addr
  length = length + 4 &,
app_X (x_disp,x_ind,x_rn,x_size) =
  if length eql 2 then
     T_lnd = x_ind;
     T_Rn = x_n;
```

りは ■ 最のののできた。■ なんの人人人 ■ EXECUTION

```
T_Size = x_Size - 1;
     T_Disp = x_disp
   ):
  if length eql 4 then
     F_Ind = x_ind;
     F_Rn = x_rn;
     F_Size = x_size - 1;
     F_Disp = x_disp
  if length eql 6 them
     S_Ind = x_ind;
     S_Rn = x_rn;
     S_Size = x_size - 1;
     S_Disp = x_disp
  length = length + 2 &,
sDS (rn,disp) =
  Src_Mode = DS;
  Src_Rn = rn;
  app_W (disp) &,
sIX (rn,x_disp,x_ind,x_rn,x_size) =
  Src_Mode = IX;
  Src_Rn = rn;
  app_X (x_disp,x_ind,x_rn,x_size) &,
sSH (addr) =
  Src Mode = SP;
  Src_Rn = SH;
  app_W (addr) &,
sLN (addr) =
  Src_Mode = SF;
  Src_Rn = LN;
  app_L (addr) &,
sPD (disp) =
  Src_Mode = SF;
  Src_Rn = FD;
  app_W (disp) &,
sPX (x_disp,x_ind,x_rn,x_size) =
  Src Mode = SP;
  Src_Rn = PX;
  app_X (x_disp,x_ind,x_rn,x_size) &,
dDS (rn,disp) =
  Dst_mode0 = DS ^ -2;
  Dist_Mode1 = DS;
  Dst_Rn = rn;
```

ステム 自然のことの名を開発できないので

■ たいかんろう 全事を入ったのでは事情によってなかな Minute かっていたい Minute かっていたのでは、Minute Minute Minu

```
app_W (disp) &,
 dlX (rn,x_disp,x_ind,x_rn,x_size) =
   Iist_mode0 = IX^{-} - 2;
   Dst_Mode1 = IX;
   Dst_Rn = rn;
   app_X (x_disp,x_ind,x_rn,x_size) &,
 dSH (addr) =
   Dst_Mode0 = SP \land -2;
   Ust Mode1 = SP;
   Dst_Rn = SH;
   app_W (addr) &,
 dLN (addr) =
   Dst_Mode0 = SP ^ -2;
   Dst_Mode1 = SP;
   Dst_Rn = LN;
   app_L (addr) &,
 dPD (disp) =
   Dst Mode0 = SP \wedge -2;
   Dst_Mode1 = SP;
   Dst_Rn = PD;
   app_W (disp) &,
 dFX (x_disp,x_ind,x_rn,x_size) ≈
   Ist_Mode0 = SP \wedge -2;
   Dst_Mode1 = SP;
   Dst Rn = FX;
   app_X (x_disp,x_ind,x_rn,x_size) %,
 adr_s (src) =
   if 1 then {s}src &.
 adr_d (dst) =
   if 1 then (d)dst &,
*!
1.8
                                                 *!
!* Instructions and their common macros in alpha-
                                                 *!
!* betical order.
1 *
abcd (r_m,Dy_Ay,Dx_Ax) =
   hux_0p = 1;
   R_M = r_m;
   Src_Rn = Dy_Ay;
   Dst_En = Dx_Ax &,
 ABCD (r_m,Dy_Ay,Dx_Ax) ==
   Opcode = 12;
```

```
abod (r_m,Dy_Ay,Dx_Ax) $ 2,
bIm (const) =
  12 = 0;
  13 = const;
  length = length + 2 &,
wIm (const) =
  12 = const \wedge -8;
  I3 = const;
  length = length + 2 %,
lIM (const) =
  I2 = const ^ -24;
  I3 = const ^ -16;
  14 = const ^ -8;
  15 = const;
  length = length + 4 &,
add (size,ea_Dn,Dn_ea) =
  Size = size;
  if 'Dn_ea eql "^DO$" or
      'Im_ea eql "^Bis" or
      'Dn_ea eql *^D2$* or
      'Dn_ea eql "^D3$" or
      'In_ea eql "^D4$" or
      'Im_ea eql "^D5$" or
      'Dn_ea eql '^D6$' or
      'Dn_ea eql *^D7$* then
      Aux_Op = 0;
      Dst_Rn = Dn_ea;
      if 'ea_lin eql "^IM.*" then
         Src_Mode = SF;
         Src_Rn = IM;
if 'size eql *^B$* then
            if 1 then {b}ea_Dn
          . };
         if 'size eql "^Ws" then
            if I then {w}ea_Dn
           Э;
         if 'size eql "^L$" then
            if 1 then {1}ea_Dn
        3
      else
        {
         adr_s (ea_In)
```

```
}
  else
     Aux_0p = 1;
    Dst_Kn = ea_Dn;
     udr_s (lin_ea)
    } &,
ADD (size,ea_Dn,Dn_ea) =
  Opcode = 13;
  add (size,ea_Dn,Dn_ea) $ &,
adda (size,ea,An) =
  Det_mode1 = 3;
  if 'size eql "^W$" then
     Aux_0p = 0
    }
  else
     Aux_0p = 1
  Dst_Rn = An;
  adr_s (ea) &,
ADDA (size,ea,An) =
  Opcode = 13;
  adda (size,ea,An) $ &,
addi (size,data,ea) =
  Size = size;
  if 'size eql "^B$" then
     12 = 0;
     I3 = data;
     length = length + 2
  if 'size eql "^W$" then
     I2 = data \wedge -B;
     I3 = data;
     length = length + 2
  if 'size eql "^L$" then
     12 = data \wedge -24;
     I3 = data \wedge -16;
     I4 = data \wedge -8;
     I5 = data;
     length = length + 4
    >:
  adr_s (ea) &,
```

```
ADDI (size,data,ea) =
   10 = 6;
   addi (size,data,ea) $ %,
 addq (size,data,ea) =
   Opcode = 5;
   Dist Rn = data;
   Size = size;
   adr s (eu) &,
ADDQ (size,data,ea) =
   Au \times Op = 0;
   addq (size,data,ea) $ %,
addx (size,r_m,Dy_Ay,Dx_Ax) =
   Sire = size;
   abcd (r_m, Hy_Ay, Hx_Ax) &,
ADDX (size,r_m,Dy_Ay,Dx_Ax) =
   Opcode = 13;
  addx (size,r_m,Dy_Ay,Dx_Ax) $ 8,
AND (size,ec_Dn,Dn_ea) =
  Opcode = 12;
  add (size,ea_Dn,Dn_ea) $ %,
ANDI (size,data,ea) =
  10 = 2;
  addi (size,data,ea) $ &,
asl (size,i_r,Dx_data,by) =
  Opcode = 14;
  Size = size;
  1_R = i_r;
  Dst_Rn = Dx_data;
  Src_Rn = Ly 2,
ASL (size,i_r,Dx_data,Dy) =
  Aux_0p = 1;
  S_R = 0;
  R_M = 0;
  asl (size, 1_r, Dx_data, Dy) $ %,
ASR (size,i_r,Dx_data,Dy) =
  Aux_0p = 0;
  S_R = 0;
  R_M = 0;
  asl (size.i_r,Dx_data,Dy) $ 8,
aslm (ea) =
  Opcode = 14;
  Dst_Mode1 = 3;
  udr_s (eu) &,
```

いと重要ないととなる。 | 「「「「「「「」」」というというできます。 | 「「「」」というというできます。 | 「「」」というというできます。

```
ASLM (eq) =
  Dst_Rn = 0;
  Aux_0p = 1;
  aslm (ea) $ &,
ASRM (ea) =
  Dst_Rn = 0;
  Aux_0p = 0;
  aslm (ea) $ &,
BB (cc,label) =
  Opcode = 6;
  Condition = cc;
  Ii = lubel $ %,
BBRA (label) =
  Opcode = 6;
  Condition = T;
  I1 = label $ &,
BBSR (label) =
  Opcode = 6;
  Condition = F;
  I1 = label $ &,
bw (label) =
  Opcode = 6;
  I1 = 0;
  12 = label ^ -8;
  I3 = label;
  length = length + 2 %,
RW(cc,label) =
  Condition = cc;
  bw (label) $ &,
BWRA (label) =
  Condition = T;
  bw (label) $ &,
BWSR (label) =
  Condition = F;
  bw (label) $ &,
bchg (s_d,Im_data,eq) =
  Opcode = 0;
  if 's_d eql "^S$"
     Dst_Rn = 4;
     Aux_0p = 0;
     13 = Im_data;
     length = length + 2
```

```
}
  else
     Dat_En = Dn_data;
     Aux_0p = 1;
  adr_s (ea) %,
RCHG (s_d,Im_data,ea) =
  Dst_Mode1 = 1;
  bchg (s_d,Im_data,ea) $ %,
BCLR (s_d,Dn_data,ea) =
  Dst_Mode1 = 2;
  bong (s_d,Dn_data,ea) $ &,
RSET (s_d,Im_data,ea) =
  Dist_Mode1 = 3;
  bchg (s_d,Dn_data,ea) $ %,
BTST (s_d,Dn_data,ea) =
  Dst_Mode1 = 0;
  being (s_d,Dn_data,ea) $ %,
chk (ea,Dn) =
  Opcode = 4;
  Dst_Rn = Dn;
  Aux_0p = 1;
  adris (ea) &,
CHK (ea,Dn) =
  Dst_Mode1 = 2;
  chk (eq, Dn) $ %,
clr (size,ea) =
  Size = size;
  adr_s (ea) &,
CLR (size,ea) =
  I0 = 0x42;
  clr (size,ea) $ &,
CmP (size,ea,Dn) =
  Opcode = 11;
  add (size,ea,Un) 4 &,
CMPA (size,ea,An) =
  Opcode = 11;
  adda (size,ea,An) $ %,
CMPI (size, data, ea) =
  10 = 12;
  addi (size,data,ea) $ %,
```

```
CMPM (size, Ay, Ax) =
    Opcode = 11;
    Aux_0p = 1;
   Src_Rn = 1;
    Size = size;
    Dst_Rn = Ax;
    Src_Rn = Ay $ %,
 db (In, label) =
   Opcode = 5;
   Dst_Mode1 = 3;
   Src_Mode = 1;
   Src_Rn = Dn;
    T2 = label ^ -8;
   I3 = label;
   length = length+ 2 %,
- DB (cc,Dn,label) =
   Condition = cc;
   db (Dn, label) $ 2,
 DBRA (Dn,label) =
   Condition = F;
   db (Dn, label) $ &,
 divs (ea,Dn) =
   Opcode = 8;
   Dst_Rn = Dn;
   Dst_Mode1 = 3;
   adr_s (ea) $ &,
 DIVS (ea,Dn) =
   Aux_Op = 1;
   divs (eu,Dn) $ %,
 DIVU (ea,Dn) =
   Aux_0p = 0;
   divs (ea,In) $ &,
 EOR (size, Dn, ea) =
   Opcode = 11;
   add (size,Dn,ea) $ %,
 EORI (size,data,ea) =
   10 = 10;
   addi (size,data,ea) $ &,
 EXG (Iix_Ax_Iy_Ay) =
   Opcode = 12;
   Dst_Rn = Dx_Ax;
   Aux_0p = 1;
   Src_Kn = Dy_Ay;
```

```
if 'I)x_Ax eql "^I)*" and
     'Dy_Ay eql "^D*" then
     Dst_Model = 1;
     Src_Mode = 0
  if 'Inx_Ax eql "^A*" and
     'Dy_Ay eql "^A*" then
     Dst_Mode1 = 1;
     Src_Mode = 1
  else
    ₹
     Dst_Mode1 = 2;
    Src_Mode = 1
    } $ &,
EXT (size, Dn) =
  Opcode = 4;
  Iist_Rn = 4;
  Aux_0p = 0;
  Size = size + 1;
  Src_Mode = 0;
  Src_Rn = Dn $ %,
.jmp(eq) =
  IO = 0 \times 4e;
  adr_s (ea) &,
JMP(ea) =
  Dst_Mode1 = 3;
  .jmp (ea) $ &,
JSR (src) =
  Dist_Mode1 = 2;
  jmp (ea) $ &,
LEA (ea, An) =
  Dst_Mode1 = 3;
  chk (ea,An) $ &,
link (An) =
  I0 = 0 \times 4d;
  Dst_Mode1 = 1;
  Src_Rn = An &,
LINK (An, disp) =
  Src_Mode = 2;
  link (An);
  I2 = disp ^ -8;
  I3 = disp;
  length = length + 2 $ %,
```

```
LSL (size,i_r,Dx_data,Dy) =
  Au \times Op = 1;
  S_R = 0;
  RM = 1;
  asl (size,i_r,Dx_data,Dy) $ 2,
LSR (size,i_r,Dx_data,Dy) =
  Aux_Op = 0;
5_R = 0;
  B_M = 1;
  asl (size,i_r,Dx_data,Dy) $ %,
LSLM (ea) =
  Dst_En = 1;
  Aux_0p = 1;
  aslm (ea) $ &,
LSRM (eq) =
  Dst_Rn = 1;
  Aux_0p = 0;
  aslm (ea) $ &,
MOVE (size,ea1,ea2) =
  if 'size eql "^B$" then
      Opcode = 1
   if 'size eql "^W$" then
      Opcode = 3
     };
   if 'size eql "^L$" then
      Opcode = 2
     ን;
   if 'eal eql "^IM.*" then
      Src_Mode = SF;
      Src_Kn = IM;
if 'size eq1 *^B$* then
        ₹
         if 1 then {b}eal
      if 'size eql "^W$" then
         if 1 then {w}ea1
      if 'size eql "^L$" then
          if 1 then {1}ea1
```

```
else
     adr_s (eal)
    };
  adr_d (ea2) $ &,
lccr (ea) =
  Opcode = 4;
  Dst_Mode0 = 0;
  Dst_Mode1 = 3;
  adr[s (ea) &,
! Move to CCR, LoaD CCR
LDCCR (eq) =
  Iist_Rn = 2;
  1ccr (ea) $ &,
! Move to SR, Loal SR
LDSR (ea) =
  list_Rn = 3;
  lccr (ea) $ &,
! Move from SR, STore SR
STSR(ea) =
  Dst_Fin = 0;
  lccr (ea) $ &,
! Move to USP, LoaD USP
LDUSP (An) =
  Src_Mode = 4;
  link (ea) $ &,
! Move from USP, Store USP
STUSP(ea) =
  Src_Mode = 5;
  link (ea) $ %,
MOVEA (size,eq,An) =
  if 'size eql "^W$" then
     Opcode = 3
     }
  else
     -{
     Opcode = 2
     >;
   Det_Rn = on;
  Dst_Mode0 = 0;
   Dst_Mode1 = 1;
   if Tea eql "^IM, *" then
      Src_Mode = SP;
      Src_Rn = IM;
```

```
if 'size eql "^W$" then
        if 1 then {w}ea
     else
        if 1 then {1}ea
  else
     adr_s (ea)
    ን $ &,
MOVEQ (data, Dn) =
  Opcode = 7;
  Dst_Rn = Un;
  Aux Op = 0;
  I1 = data $ 8,
muls (ea, Dn) =
  Opcode = 12;
  Dst_Rn = Dn;
  Ist_Mode1 = 3;
  adr_s (ea) &,
MULS (ea, Dn) =
  Aux_0p = 1
  muls (ea,Im) $ 2,
MULU (ea, Dn) =
  hux_0p = 0;
  muls (ea,Dn) $ &,
NBCD (ea) =
  10 = 0 \times 48;
  Dst_Mode1 = 0;
  acr_s (ea) $ &,
NEG (size,ea) =
  IO = 0x44;
  clr (size,ea) $ &,
NEGX (size,ea) =
  IO = 0x40;
  clr (size,eu) $ &,
nop =
  I0 = 0 \times 4e;
  Dst_mode1 = 1;
  Src_Mode = 6 %,
NOP =
```

```
Src_Rn = 1;
  nop $ &,
NOT (size,eq) =
  IO = 0x46;
  clr (size,ea) $ %,
OR (size,ea_Dn,Dn_ea) =
  Opcode = 8;
  add (size,ea_Dn,Dn_ea) $ %,
ORI (size, data, ea) =
  10 = 0;
  addi (size,data,ea) 4 %,
PEA (src) =
  10 = 0 \times 48;
  Dst_mode = 1;
  adr_s (src) $ &,
RESET =
  Src_Rn = 0;
  nop $ &,
ROL (size,i_r,Dx_data,Dy) =
  \Delta u \times 0 = 1;
  S_R = 1;
  R M = 1;
  asl (size,i_r,Dx_data,Dy) * &,
ROE (size,i_r,Dx_data,Dy) =
  Aux_0p = 0;
  S_{R} = 1;
  R_M = 1;
  asl (size,i_r,Dx_data,Dy) $ %,
ROLM (ea) =
  Dst_Rn = 3;
  Aux_0p = 1;
  aslm (ea) $ &,
RDRM(ea) =
  Dst_Rn = 3;
  Aux_0p = 0;
  aslm (ea) $ %,
ROXL (size, i_r, Dx_data, Dy) =
  Au \times Op = 1;
  S_R = 1;
  R_M = 0;
  asl (size,i_r,Dx_data,Dy) $ %,
ROXR (size,i_r,Dx_data,Dy) =
```

```
Aux_Op = 0;
 SR = 1;
 R_M = 0;
 asl (size,_r,bx_data,Dy) $ &,
ROXLM (ea) =
  Dst_Bn = 2;
  Aux_Op = 1;
  aslm (ea) $ %,
ROXRM (ea) =
 Dst_En = 2;
  Aux_0p = 0;
  aslm (ea) $ &,
RTE =
  Src_{Rn} = 3;
  nop $ &,
RTR =
  Src_Rn = 7;
  nop $ &,
RTS =
  Src_En = 5;
  nop $ &,
SECH (r_m,Dy_Ay,Dx_Ax) =
  Opcode = 8;
  abod (r_m,Dy_Ay,Dx_Ax) $ &,
S (cc,ea) =
  Opcode = 5;
  Condition = cc;
  Dst_Mode1 = 3;
  adr_s (ea) $ &,
STOP (data) =
  Src_Rn = 2;
  nop;
  12 = data ^ -8;
  I3 = data;
  length = length + 2 $ %,
SUB (size,ea_Dn,Dn_ea) =
  Opcode = 9;
  add (size,ea_Dn,Dn_ea) $ &,
SUBA (size,eq,An) =
  Opcode = 9;
  adda (size,ea,An) $ &,
SUBI (size,data,ea) =
```

```
10 = 8;
  addi (size,data,ea) $ %,
SUBQ (size,data,ea) =
  Aux_0p = 1;
  addq (size,data,ea) $ &,
 SUBX (size, r_m, Dy_Ay, Dx_Ax) =
  Opcode = 9;
  addx (size,r_m,Dy_Ay,Dx_Ax) $ &,
 SWAP (Dn) =
  10 = 0 \times 48;
  Dst_Mode1 = 1;
  Src_Mode = 0;
  Src_Rn = Dn $ &,
 TAS (ea) =
   I0 = 0 \times 4a;
  Dst_Mode1 = 3;
   udr_s (eu) % &,
 TRAP (vector) =
   IO = 0x4d;
   Dst_Mode1 = 1;
   I_R = 0;
   S_R = 0;
   R_M = vector ^ -3;
   Src_Rn = vector $ &,
 TRAPV =
   Src_Rn = 6;
   nop $ &,
 TST (size,ea) =
   TO = 0 \times 4a;
   clr (size,ea) $ &,
 UNLK (An) =
   Src_Mode = 3;
   link (An) $ &,
*!
! *
                                                 *!
!* Psuedo instructions for constants.
                                                 *!
1 %
CB (const) =
   10 = const;
   length = 1 $ &,
 CW (const) =
```

```
I0 = const ^ -8;
I1 = const $ 2,

CL (const) =
    I0 = const ^ -24;
I1 = const ^ -16;
I2 = const ^ -8;
I3 = const;
length = 4 $ $$
```

(

Appendix E: MC68000 Linking/Loader Description

```
! *
                                        *!
!* m68000.i.
                                        *!
                                        *!
!* Linking Loader description for Motorola 68000
                                        *!
!* microprocessor.
!* No input requirements.
                                        *!
!* Generates mó8000.a on compilation.
                                        *!
                                        *!
!* Use "inter" to compile.
                                        *!
! *
                                        *!
!* Author
         : Samir S. Shah.
         : Fall 1979.
                                        *!
!* Date
!* Modified : Samir S. Shah.
         : Spring 1981.
                                        *!
!* Date
                                        *!
*!
                                        *!
!* Declaration for instruction lengths and widths.
!* memory is byte addressable, default instruction
                                        *!
                                        *!
!* length is 2 bytes, maximum instruction length is
                                        *!
!* 10 bytes.
                                        * 1
1 *
instr
 ID10,23<8>$
format
1 *
!* Fields of instruction bytes zero and one.
                                        * 1
                                        *!
! *
= 1003<7:4>,
 Opcode
 Condition = 1003<3:0>,
 Dst_Rn
         = 1003(3:1),
 Ist_Mode0 = IE03<0>,
         = I[0]<0>,
 hung Op.
 Det model = I[1]\langle 7:6\rangle,
        = IE13<7:6>,
 Size
 Src_Mode = I[1]<5:3>,
         = IC13<5>,
 I_{\perp}R
         = I[1]<4>,
 S_R
```

アンドラ こくじゅつ サインスタング いんかき かんきん ちゅう おおおく にんしんしゅう

```
= I(1)<3>.
 F._M
 Src Bn
       = I[13<2:0>,
! *
                                   *!
!* Fields of instruction bytes Two and three.
                                   *!
1 *
T_1nd = 1023<7>,
 TERN
     = I[2]<6:4>,
 T_Size = I[23<3>,
 T Disp = I[3]<7:0>,
! *
                                   x 1
                                   *!
!* Fields of instruction bytes Four and five.
                                   * 1
1 *
= I[4]<7>,
 F_Ind
 F Bn
     = I[4]<6:4>,
  _Size = I[4]<3>,
 F Disp = 1053<7:0>,
丰米
                                   *!
                                   *!
!* Fields of instruction bytes Six and seven.
! *
                                   *!
S Ind
     = 1[6]<7>,
 S Rn
     = I[6](6:4),
 S_Size = I[6]<3>,
 S_Disp = I[7]\langle 7:0 \rangle,
! ж
                                   *!
!* Shorter names for instruction bytes.
                                   *!
1 🖈
                                   *!
IO = IEOJ \langle 7:O\rangle,
 11 = 1013 < 7:0 >,
 12 = I[2]<7:0>,
 13 = 1033 < 7:0 >
 14 = I[4]<7:0>,
 15 = IE5J<7:0>,
 16 = I[6]<7:0>,
 17 = 1073 < 7:0 >,
 18 = I(8)(7:0),
 19 = 1093<7:0>$
```

■ まじょうから ■ あんこうから ■ なっかいかつ ■ あんらん

```
space
  <0:32767>$
transfer
  Knew
   IO ≈ 0×4e $
   II = 0xf9 $
   12 = address ^ -24 $
   I3 = address ^ -16 $
   14 = address ^ -8 $
  15 = address $
mode
 case (labelent eql 2) and
       (length eql 10):
    I2 = address[1] ^ -24 $
    I3 = address[1] \land -16 $
    14 = address[1] ^ -8 $
    IS = address[1] $
    16 = address[5] ^ -24 $
    17 = address[5] ^ -16 $
    18 = address[5] ^ -8 $
    19 = address[5] $
  break$
  esac,
 case Opcode eql 6:
       I1 = address - . - 2$
       break$
  esuc,
 default:
 esac$
```

Appendix F: Simulation Test Routines

This appendix identifies the Metamicro test routines processed for each of the MC68000 instruction and exception models that were simulated. They are:

Instruction Routine	Page
1. MOVE.W D1,D2	F-3
2. MOVE.W D1, (A1)	F-3
3. MOVE.L Dl,Al	F-3
4. MOVE.W D1, (A1) +	F-3
5. MOVE.W D1,04(A1)	F-4
6. MOVE.W D1,04(A1,D7)	F-4
7. MOVE.W D1,\$2004	F-4
8. MOVE.W Al, D3	F-5
9. MOVE.W (A1),D2	F-5
10. MOVE.W (A1)+,D6	F-5
11. MOVE.W -(A1),D4	F-5
12. MOVE.W 04(A1),D1	F-6
13. MOVE.W 04(A1,D7),D2	F-6
14. MOVE.W \$2004,D5	F-6
15. MOVE.W \$2004,\$2008	F-7
16. MOVE.W #\$5555,D1	F-7
17. ADD.W D3,D5	F-7
18. BEQ START	F-8
19. BTST D1,(A1)	F-8
20. Illegal Instruction Exception	F-8

21. Address Error Exception

F-8

```
1. MOVE.W D1,D2
include mc68000.m$
begin
                             ! Move contents of data register
  MOVE (W, DR(D1), DR(D2))
  MOVE (W,DR(D1),DR(D2))
                             ! Dl to data register D2
  MOVE (W,DR(D1),DR(D2))
  MOVE (W,DR(D1),DR(D2))
  MOVE (W, DR(D1), DR(D2))
  MOVE (W,DR(D1),DR(D2))
                             ! Jump to address pointed to by
  JMP (IR(A0))
                             ! address register A0
end
2. MOVE.W D1, (A1)
include mc68000.m$
begin
                             ! Move contents of data register
  MOVE (W, DR(D1), IR(A1))
  MOVE (W,DR(D1), IR(A1))
                             ! D1 to memory location pointed
                             ! to by address register Al
  MOVE (W, DR(D1), IR(A1))
  MOVE (W,DR(D1),IR(A1))
  MOVE (W, DR(D1), IR(A1))
  MOVE (W, DR(D1), IR(A1))
                             ! Jump to address pointed to by
  JMP
       (IR(A0))
                             ! address register A0
end
3. MOVE.L D1,A1
include mc68000.m$
begin
                             ! Load routine at address 1000 hex
      = 0x1000$
                              ! Set supervisor mode bit to zero
  ANDI (W, 0xdfff, DR(SR))
                              ! (user mode)
                             ! Move 32-bit contents of data
  MOVE (L, DR(D1), AR(A1))
                             ! register D1 to address register
  MOVE (L,DR(D1),AR(A1))
                             ! Jump to address pointed to by
  JMP
       (IR(AG))
                              ! address register A0
end
4. MOVE.W D1, (A1) +
include #c68000.m$
begin
                              ! Load routine at address 1000 hex
      = 0x1000$
                              ! Set supervisor mode bit to zero
  ANDI (W, 0xdfff, DR(SR))
                             ! (user mode)
                             ! Move contents of data
  MOVE (W,DR(D1),AI(A1))
```

```
MOVE (W, DR(D1), AI(A1))
                             ! register D1 to memory location
                             ! pointed to by address register Al
                             ! and then increment Al by 2
  MOVE (L,DR(D2),AR(A1))
                             ! Re-initialize address register Al
  JMP
       (IR(A0))
                             ! Jump to address pointed to by
                             ! address register A0
end
5. MOVE.W D1,04(A1)
include mc68000.m$
begin
      = 0x1000$
                             ! Load routine at address 1000 hex
                             ! Set supervisor mode bit to zero
  ANDI (W, 0xdfff, DR(SR))
                             ! (user mode)
  MOVE (W, DR(D1), DS(A1,4))
                             ! Add displacement (4) to address
                             ! register Al to form address
                             ! which will receive the contents
                             ! of data register Dl
  MOVE (W, DR(D1), DS(A1,8))
                             ! Repeat with displacement 8
                             ! Jump to address pointed to by
  JMP
       (IR(A0))
                             ! address register A0
end
6. MOVE.W D1,04(A1,D7)
include mc68000.m$
begin
      = 0x1000$
                             ! Load routine at address 1000 hex
  ANDI (W, 0xdfff, DR(SR))
                             ! Set supervisor mode bit to zero
                             ! (user mode)
  MOVE (W,DR(D1),IX(A1,4,0,D7,1)) ! Sum displacement (4), the
                             ! contents of data register D7 and
                             ! address register Al to form ad-
                             ! dress of memory location to
                             ! receive data register Dl
  MOVE (W, DR(D1), IX(A1,4,0,D7,1)) ! Repeat with displacement 8
  JMP
       (IR(A0))
                             ! Jump to address pointed to by
                             ! address register A0
end
7. MOVE.W D1, $2004
include mc68000.m$
begin
      = 0x1000$
                             ! Load routine at address 1000 hex
 ANDI (W, 0xdfff, DR(SR))
                             ! Set supervisor mode bit to zero
                             ! (user mode)
 MOVE (W,DR(D1),SH(0x2004))! Move contents of data register
                             ! D1 to memory location 2004 hex
 MOVE (W, DR(D1), SH(0x2008))! Repeat at location 2008 hex
 JMP
       (IR(A0))
                             ! Jump to address pointed to by
```

ことは自動を大きたとして、一気のないないない

```
! address register A0
end
8. MOVE.W Al, D3
include mc68000.m$
begin
                             ! Load routine at address 1000 hex
      = 0x1000$
                             ! Set supervisor mode bit to zero
  ANDI (W, 0xdfff, DR(SR))
                             ! (user mode)
  MOVE (W, AR(Al), DR(D3))
                             ! Move contents of address register
                             ! Al to data register D3
  MOVE (W, AR(Al), DR(D3))
                             ! Jump to address pointed to by
  JMP (IR(A0))
                             ! address register A0
end
9. MOVE.W (A1),D2
include mc68000.m$
begin
                             ! Load routine at address 1000 hex
      = 0x1000$
                             ! Set supervisor mode bit to zero
  ANDI (W, 0xdfff, DR(SR))
                             ! (user mode)
 MOVE (W, IR(Al), DR(D2))
                             ! Move contents of memory location
                             ! pointed to by address register
                             ! Al to data register D2
  MOVE (W, IR(Al), DR(D2))
                             ! Jump to address pointed to by
  JMP (IR(A0))
                             ! address register A0
end
10. MOVE.W (A1) + D6
include mc68000.m$
begin
                             ! Load routine at address 1000 hex
      = 0x1000$
                             ! Set supervisor mode bit to zero
  ANDI (W, 0xdfff, DR(SR))
                             ! (user mode)
                             ! Move contents of memory location
  MOVE (W,AI(A1),DR(D6))
                             ! pointed to by address register
  MOVE (W,AI(A1),DR(D6))
                             ! Al to data register D6
                             ! then increment Al by 2
  JMP
       (IR(A0))
                             ! Jump to address pointed to by
                             ! address register A0
end
11. MOVE.W -(A1),D4
include mc68000.m$
begin
      = 0x1000$
                             ! Load routine at address 1000 hex
  ANDI (W, 0xdfff, DR(SR))
                            ! Set supervisor mode bit to zero
```

```
! (user mode)
  MOVE (W,AD(A1),DR(D4))
                             ! Decrement address register Al
                             ! by 2 and then use to identify
                             ! memory location to receive con-
                             ! tents of data register D4
                             ! Repeat for data register D3
  MOVE (W, AD(A1), DR(D3))
  MOVE (L,DR(D2),AR(A1))
                             ! Reset Al
                             ! Jump to address pointed to by
  JMP
       (IR(A0))
                             ! address register A0
end
12. MOVE.W 04(A1),D1
include mc68000.m$
begin
                             ! Load routine at address 1000 hex
     = 0x1000$
                             ! Set supervisor mode bit to zero
  ANDI (W, 0xdfff, DR(SR))
                             ! (user mode)
                             ! Move contents of memory location
  MOVE (W,DS(A1,4),DR(D1))
                             ! determined by summing displace-
                             ! ment 4 and contents of address
                             ! register Al-to data register Dl
                             ! Repeat with displacement 8 and
 MOVE (W, DS(A1,8), DR(D2))
                             ! data register D2
                             ! Jump to address pointed to by
  JMP
       (IR(A0))
                             ! address register A0
end
13. MOVE.W 04(Al,D7),D2
include mc68000.m$
begin
                             ! Load routine at address 1000 hex
       = 0x1000$
                             ! Set supervisor mode bit to zero
  ANDI (W, 0xdfff, DR(SR))
                             ! (user mode)
  MOVE (W,IX(Al,4,0,D7,1),DR(D2)) ! Move contents of memory
                             ! location determined by the sum of
                             ! displacement 4, contents of data
                             ! remister D7, and address register
                             ! Al - to data register D2
  MOVE (W, IX(A1,4,0,D7,1),DR(D3)) ! Repeat for data register D3
                             ! Jump to address pointed to by
  JMP
       (IR(A0))
                             ! address register A0
end
14. MOVE.W $2004,D5
include mc68000.m$
begin
                             ! Load routine at address 1000 hex
      \approx 0 \times 1000 $
                             ! Set supervisor mode bit to zero
  ANDI (W, 0xdfff, DR(SR))
                             ! (user mode)
```

```
MOVE (W,SH(0x2004),DR(D5)) ! Move word at memory location
                             ! 2004 hex into data register D5
  MOVE (W,SH(0x2004),DR(D6)) ! Repeat for data register D6
  JMP (IR(A0))
                             ! Jump to address pointed to by
                             ! address register AO
end
15. MOVE.W $2004,$2008
include mc68000.m$
begin
      = 0x1000s 
                             ! Load routine at address 1000 hex
  ANDI (W, 0xdfff, DR(SR))
                             ! Set supervisor mode bit to zero
                             ! (user mode)
  MOVE (W,LN(0x2004),LN(0x2008)) ! Move word at memory location
                             ! 2004 hex into memory location
                             ! 2008 hex
  JMP
       (IR(A0))
                             ! Jump to address pointed to by
                             ! address register A0
end
16. MOVE.W #$5555,D1
include mc68000.m$
begin
     = 0x1000$
                             ! Load routine at address 1000 hex
  ANDI (W, 0xdfff, DR(SR))
                             ! Set supervisor mode bit to zero
                             ! (user mode)
 MOVE (W, IM(0x5555), DR(D1)) ! Move data word 5555 hex into
 MOVE (W, IM(0x5555), DR(Dl)) ! data register Dl
  JMP
       (IR(A0))
                             ! Jump to address pointed to by
                             ! address register A0
end
17. ADD.W D3,D5
include mc68000.m$
begin
     = 0x1000$
                             ! Load routine at address 1000 hex
 ANDI (W, 0xdfff, DR(SR))
                             ! Set supervisor mode bit to zero
                             ! (user mode)
 MOVE (W,DR(D1),DR(D5))
                             ! Move contents of data register Dl
                             ! into data register D5
 ADD
       (W, DR(D3), D5)
                             ! Sum contents of data registers D3
                             ! and D5 and store in D5
 MOVE (W,DR(D5),IR(A2))
                             ! Move result to memory location
                             ! identified by address register A2
                             ! Jump to address pointed to by
 JMP
       (IR(A0))
                             ! address register AO
```

end

```
18. BEQ START
include mc68000.ms
begin
  START:
          MOVE (W,DR(D1),DR(D3)) ! This move will clear Zero
                                   ! condition code of SR
          BB
                (EQ, START)
                                   ! Branch not taken
          MOVE
                (W, DR(D2), DR(D3)) ! Set Zero condition code
          BB
                                   ! Branch taken
                (EQ,START)
           JMP
                (IR(A0))
                                   ! Jump to address pointed to
                                   ! by address register A0
end
19. BTST D1, (A1)
include mc68000.ms
begin
  MOVE (W, DR(D2), DR(D3)) ! Fill prefetch queue
  BTST (R,Dl,IR(Al))
                          ! Test bit identified by contents
                          ! of data register Dl in memory
                          ! location identified by address
                          ! register Al
  MOVE (W, DR(D2), DR(D3)) ! To determine completion of BTST
  JMP
                             ! Jump to address pointed to by
       (IR(A0))
                             ! address register A0
end
20. Illegal Instruction Exception
include mc68000.m$
begin
      = 0x1000$
                             ! Load routine at address 1000 hex
  ANDI (W, 0xdfff, DR(SR))
                             ! Set supervisor mode bit to zero
                             ! (user mode)
  MOVE (W,DR(D1),DR(D2))
                             ! Surround illegal instruction with
  MOVE (W,DR(D1),DR(D2))
                             ! MOVE's to fill prefetch queue
  MOVE (W, DR(D1), DR(D2))
                             ! and isolate illegal instruction
  CW
       (0x4afc)
                             ! Illegal Instruction
  MOVE (W,DR(D1),DR(D2))
  JMP
       (IR(A0))
                             ! Jump to address pointed to by
                             ! address register A0
end
21. Address Error Exception
include mc68000.m$
begin
       = 0x1000$
                             ! Load routine at address 1000 hex
 ANDI (W, 0xdfff, DR(SR))
                             ! Set supervisor mode bit to zero
                             ! (user mode)
 MOVE (W,DR(D1),IR(A1))
                             ! Attempt to move word from data
```

! register D1 to odd address con-

! tained in address register Al NOP ! This instruction will be pre-! fetched during address error ! instruction ! Jump to address pointed to by ! address register A0 JMP (IR(A0)) = 0x2040\$! Load exception handler routine ! at memory location 2040 hex ! Pop system stack to retrieve ADDQ (L,4,AR(A7))ADDQ (L,4,AR(A7))! program counter pointing to ! JMP (A0) instruction RTE ! Return from Exception end

Appendix G: Simulation Control Files

This appendix contains the global files used to build each simulation and output the data of interest.

Topology File

1.1.2 ■行われたのでは、「Manager Manager Man

The following is an overlay of the topology files used to identify and bind each of the simulation's components for processing by N.mPc's Ecologist:

signal	DBUS(16), ABUS(23);	! Data Bus ! Address Bus
processor	cpu = "root.sim"	! The file "root.sim" ! contains the compiled ! MC68000 model
time delay	1000000 ns;	! A simulation delay ! corresponds to 1 ms
connections	DBUS = DBUS,	! Connect processor's ! Data Bus pins to ! Data Bus
	ABUS = ABUS;	! Ditto Address Bus Pins
initial	M = ROOT;	! Load memory "M" with ! executable code contained ! in the file "ROOT"

This file will support the modeling of any of the instructions if the file "root.sim" is replaced with the compiled version of the instruction's hardware component in the "processor" declaration, and the name of the file containing the executable code for the instruction's test routine is substituted for the file "ROOT" in the "initial" declaration.

Simulate

The file "simulate" governed each simulation by determining when simulation breakpoints would occur and then executing the appropriate signal output file to display the desired data. It appears as follows:

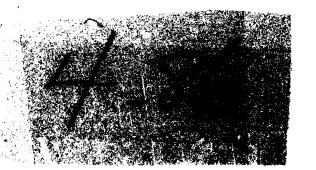
Signals1/Signals2

Signals1 and Signals2 were executed at the appropriate simulation breakpoints to accomplish the actual data display. These two files differ only in the clock phase signal displayed. Signals1 will display the state of PHI1 as a part of its output while Signals2 displays PHI2. They both consist of the following Runtime statements:

```
base 10
                    ! Want clock cycle displayed in base 10
examine :T
                   ! Display clock cycle
examine :PHIl
                   ! Display phase 1 of current clock cycle
                   ! Signals2 will display phase 2
base 2
                   ! Display remaining data in base 2
examine :FC
examine :FC ! Display Function Code Signals examine :DTACKN ! Display Data Transfer Acknowledge
examine :RW
                   ! Display Read/Write
examine :LDSN
                  ! Display Lower Data Strobe
                  ! Display Upper Data Strobe
examine :UDSN
                  ! Display Address Strobe
examine :ASN
examine :DBUS ! Display state of Data Bus
```

END

FILMED



DTIC